Distributed Receding Horizon Control of Cost Coupled Systems

William B. Dunbar

Abstract— **This paper considers the problem of distributed control of dynamically decoupled systems that are subject to decoupled constraints, while their states are coupled non-separably in the cost function of an optimal control problem. A distributed receding horizon control (RHC) algorithm is presented, in which each subsystem (agent) computes its own control locally. The implementation presumes synchronous parallel updates. Coupled agents, referred to as neighbors, coordinate by the exchange of an assumed state trajectory prior to each update. The algorithm is an improvement over the previous algorithm developed by the author, in that stability is guaranteed without adding any additional constraints to each local optimal control problem. Instead, a move suppression term is added into each local cost function, which penalizes the deviation of the computed state trajectory from the assumed state trajectory. Closed-loop stability follows if the weight on the move suppression term is larger than a parameter that bounds the amount of coupling in the cost function between neighbors.**

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

I. INTRODUCTION

This paper considers the problem of controlling of a set of dynamically decoupled agents that are required to perform a cooperative task. An example of such a situation is a group of autonomous vehicles cooperatively converging to a desired formation, as explored in [5], [12]. One control approach that accommodates a general cooperative objective is receding horizon control (RHC). In RHC, the current control action is determined by solving a finite horizon optimal control problem at each sampling instant. In continuous time formulations, each optimization yields an open-loop control trajectory and the initial portion of the trajectory is applied to the system until the next sampling instant. A cooperative objective can be incorporated into RHC by appropriate choice of the cost function in the optimal control problem. Agents that are coupled in the cost function are referred to as neighbors.

A *distributed implementation* of RHC is here presented in which each agent is assigned its own optimal control problem, optimizes only for its own control at each update, and coordinates with neighboring agents. Neighbors coordinate by the exchange of an assumed state trajectory prior to each update. The work presented here is a continuation of [5], in which a consistency constraint is included in each local problem to ensure agents do not deviate too far from the assumed trajectory. The implementation here is an improvement, in that the consistency constraint is no longer necessary. Instead, a move suppression term is added into each local cost function, which penalizes the deviation of the computed state trajectory from the assumed state trajectory. Closed-loop stability follows if the weight on the move suppression term is larger than a parameter that bounds the amount of coupling in the cost function between neighbors. While move suppression terms are traditionally on the rate of change of the control inputs in discrete-time applications of model predictive process control, the move suppression term here involves the state trajectory.

In the context of multiple autonomous vehicle missions, several researchers have proposed hierarchical/leader-follower distributed RHC schemes [8], [13], [14]. In some of these approaches, coupling constraints are admissible, which is an advantage. In contrast to hierarchical methods, the distributed RHC framework here uses no hierarchical assignment and agents compute their own control in parallel. Another recent work in which agents update in parallel and inter-agent communication delay is admitted is [6], although a conservative small-gain condition is required for stability. The paper is organized as follows. Section II begins by defining the agent dynamics and constraints, and the generic form of coupling cost function. In Section III, distributed optimal control problems are defined for each agent, and the distributed RHC algorithm is stated. Feasibility and stability results are then given in Section IV, and Section V provides conclusions.

II. SYSTEM DESCRIPTION AND OBJECTIVE

In this section, the system dynamics and control objective are defined. We make use of the following notation. The symbol $\mathbb{R}_+$ represents the set of non-negative reals. The symbol $\|\cdot\|$ denotes the Euclidean 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, given 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 $t \in \mathbb{R}$.

In the control problem below, subsystems will be coupled in an integrated cost function of an optimal control problem. For example, vehicles i and j might be coupled in the integrated cost by the term $\|q_i - q_j + d_{ij}\|^2$, where $q_{(\cdot)}$ is the position of the vehicle, and d_{ij} is a constant desired relative position vector that points from i to j. The purpose of the distributed RHC approach is to decompose the overall cost so that, in this example, i and j would each take a fraction of the term $\|q_i - q_j + d_{ij}\|^2$ (among other terms)

Research supported in part by NIH grant #K25 HG004035-01.

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

in defining their local cost functions. Then, i and j update their RHC controllers *in parallel*, exchanging information about each others anticipated position trajectory so that each local cost can be calculated. More generally, the coupling cost terms may not be quadratic; the assumption is that the coupled cost terms are *non-separable*, so that agents i and j must exchange trajectory information if they are coupled via the cost function. An example of quadratic coupling cost functions is examined in this paper, while non-quadratic coupling cost functions are treated elsewhere [4]. The concept of non-separability is now formally defined.

A non-negative function $g : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ is called *non-separable* in $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ if g is *not additively separable* for all $x, y \in \mathbb{R}^n$. That is, g cannot be written as the sum of two non-negative functions $g_1 : \mathbb{R}^n \to \mathbb{R}_+$ and $g_2 : \mathbb{R}^n \to \mathbb{R}_+$ such that $g(x, y) = g_1(x) + g_2(y)$ for all $x, y \in \mathbb{R}^n$. By this definition, note that g is non-separable in vectors x and y even if only one of the components of y is coupled non-separably to any of the components of x.

The objective is to stabilize a collection of subsystems, referred to as *agents*, to an equilibrium point using RHC. In addition, each agent is required to *cooperate* with a set of other agents, where cooperation refers to the fact that every agent has incentive to optimize the collective cost function that couples their state to the states of other agents. For each agent $i \in \mathcal{V} \triangleq \{1, ..., N_a\}$, 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 decoupled, time-invariant nonlinear system dynamics are given by

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

While the system dynamics can be different for each agent, the dimension of every agents state (control) is assumed to be the same, for notational simplicity and without loss of generality. Each agent i is also subject to the decoupled state and input constraints,

$$z_i(t) \in \mathcal{Z}, \quad u_i(t) \in \mathcal{U}, \quad \forall t \geq t_0,$$

where $\mathcal{Z}$ and $\mathcal{U}$ are also assumed to be common to every agent i for notational simplicity and without loss of generality. The cartesian product is denoted $\mathcal{Z}^{N_a} = \mathcal{Z} \times \cdots \times \mathcal{Z}$. The concatenated vectors are denoted $z = (z_1, ..., z_{N_a})$ and $u = (u_1, ..., u_{N_a})$. In concatenated vector form, the system dynamics are

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

where $f(z, u) = (f_1(z_1, u_1), ..., f_{N_a}(z_{N_a}, u_{N_a}))$. The desired equilibrium point is the origin, and some standard assumptions regarding the system are now stated

Assumption 1: The following holds for every $i \in \mathcal{V}$:

- **(a)** $f_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuous, $0 = f_i(0, 0)$, and f_i is locally Lipschitz in z_i;
- **(b)** $\mathcal{Z}$ is a closed connected subset of $\mathbb{R}^n$ containing the origin in its interior;
- **(c)** $\mathcal{U}$ is a compact, convex subset of $\mathbb{R}^m$ containing the origin in its interior;
- **(d)** every state trajectory z_i is bounded, satisfying $\|z_i(t)\| \leq \rho$ for some $\rho \in (0, \infty)$ and for all $t \geq t_0$.

That states remain bounded is a prerequisite for any implemented nonlinear optimization algorithm [11]. It is assumed that a single collective cost function $L : \mathbb{R}^{nN_a} \to \mathbb{R}_+$ is provided that couples the states of the agents, and that each agent has incentive to minimize this function with respect to their own state. For each $i \in \mathcal{V}$, let $\mathcal{N}_i \subseteq \mathcal{V} \setminus \{i\}$ be the set of other agents whose states are coupled non-separably to z_i in $L(z)$. By definition, $j \in \mathcal{N}_i$ if and only if $i \in \mathcal{N}_j$, for all distinct $i, j \in \mathcal{V}$. Denote $N_i = |\mathcal{N}_i|$ and let $z_{-i} = (z_{j_1}, ..., z_{j_{N_i}})$ be the collective vector of states coupled non-separably to z_i in $L(z)$.

Assumption 2: The function $L : \mathbb{R}^{nN_a} \to \mathbb{R}_+$ is continuous, positive definite and can be decomposed as follows: for every $i \in \mathcal{V}$ there exists an integer $N_i \in \{1, ..., N_a - 1\}$ and a continuous and non-negative function $L_i : \mathbb{R}^n \times \mathbb{R}^{nN_i} \to \mathbb{R}_+$, not identically zero, such that

- **(a)** $L_i(z_i, z_{-i})$ is non-separable in $z_i \in \mathbb{R}^n$, $z_{-i} \in \mathbb{R}^{nN_i}$;
- **(b)** $\sum_{i \in \mathcal{V}} L_i(z_i, z_{-i}) = L(z)$;
- **(c)** there exists a positive constant $c_i \in (0, \infty)$ such that

$$L_i(x, y) \leq L_i(x, w) + c_i \|y - w\|,$$

  for all $x \in \mathbb{R}^n$ and for all $y, w \in \mathbb{R}^{nN_i}$. The constant c_i is referred to as the **strength-of-coupling parameter**, and the term $c_i \| \cdot \|$ is referred to as the **cost coupling bound**.

Observe that N_i is the *number of neighbors* for each $i \in \mathcal{V}$. The example coupling cost function L below, is shown to satisfy the structure required in Assumption 2.

Example 1 (Quadratic Coupling Cost): Let $L(z) = \|z\|_Q^2$, where $Q = Q^T > 0$ is full rank. It is straightforward to identify $Q_i = Q_i^T \geq 0$ such that with $L_i(z_i, z_{-i}) = \|(z_i, z_{-i})\|_{Q_i}^2$ Assumptions 2(a)-(b) are satisfied. The expression for c_i is next derived to satisfy Assumption 2 (c). For any $x \in \mathbb{R}^n$ and $y, w \in \mathbb{R}^{nN_i}$,

$$\begin{aligned} L_i(x, y) - L_i(x, w) &= \left\| \begin{matrix} x \\ y \end{matrix} \right\|_{Q_i}^2 - \left\| \begin{matrix} x \\ w \end{matrix} \right\|_{Q_i}^2 \\ &= (y - w)^T Q_{i,c}(y + w) + 2x^T Q_{i,b}(y - w), \\ &\text{with } Q_i = \begin{bmatrix} Q_{i,a} & Q_{i,b} \\ Q_{i,b}^T & Q_{i,c} \end{bmatrix}. \end{aligned}$$

Since every state z_i is assumed to satsify $\|z_i\| \leq \rho$,

$$\begin{aligned} &(y - w)^T Q_{i,c}(y + w) + 2x^T Q_{i,b}(y - w) \\ &\quad \leq \|y - w\| 2\rho \left\{ \lambda_{\max}(Q_{i,c}) N_i + \lambda_{\max}^{1/2}(Q_{i,b}^T Q_{i,b}) \right\}. \end{aligned}$$

It is straightforward to show that $\lambda_{\max}(Q_i) \geq \lambda_{\max}^{1/2}(Q_{i,b}^T Q_{i,b})$ and $\lambda_{\max}(Q_i) \geq \lambda_{\max}(Q_{i,c})$ [4]. Thus,

$$L_i(x, y) - L_i(x, w) \leq c_i \|y - w\|,$$

with $c_i = 2\rho \lambda_{\max}(Q_i)(N_i + 1)$. □

Other examples of costs that satisfy Assumption 2, including quadratic and non-quadratic multi-vehicle formation costs, are provided in [4]. If the cost L (and L_i) are norm costs, instead of norm-squared (quadratic) costs, it is trivial to identify c_i using the triangle inequality. Another observation, related to the form of move suppression used later, is that norm-squared (quadratic) costs do not satisfy $L_i(x,y) \leq L_i(x,w) + c_i\|y-w\|^2$ for all x, y, w, for any given positive constant c_i.

III. 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. The receding horizon update times are denoted $t_k = t_0 + \delta k$, 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^{th}-update.

At each update, every agent optimizes only for its own open-loop control, given its current state and that of its neighbors. Since each cost $L_i(z_i, z_{-i})$ depends upon the neighboring states z_{-i}, each agent i must 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 from each neighbor. Likewise, agent i transmits an assumed state to all neighbors prior to each optimization. To distinguish the different trajectories, the following notation is used for each agent $i \in \mathcal{V}$:

$$\begin{aligned} z_i(t) \quad &: \quad \text{the actual state, at any time } t \geq t_0, \\ z_i^p(\tau;t_k) \quad &: \quad \text{the predicted state, } \tau \in [t_k, t_k+T], \\ \hat{z}_i(\tau;t_k) \quad &: \quad \text{the assumed state, } \tau \in [t_k, t_k+T], \end{aligned}$$

for any $k \in \mathbb{N}$. For the RHC implementation here, $z_i(t) = z_i^p(t;t_k)$ for all $t \in [t_k, t_{k+1}]$ and any $i \in \mathcal{V}$. The predicted and assumed control trajectories are likewise denoted $u_i^p(\tau;t_k)$ and $\hat{u}_i(\tau;t_k)$, respectively. Let $\hat{z}_{-i}(\tau;t_k)$ be the vector of assumed state trajectories of the neighbors of i, corresponding to current time t_k. At time t_k, the cost function $J_i(z_i(t_k), u_i^p(\cdot;t_k))$ for the optimal control problem for each agent $i \in \mathcal{V}$ is

$$\int_{t_k}^{t_k+T} \Big[L_i(z_i^p(s;t_k), \hat{z}_{-i}(s;t_k)) + \|u_i^p(s;t_k))\|_{R_i}^2 + b_i\|z_i^p(s;t_k) - \hat{z}_i(s;t_k)\| \Big] \mathrm{d}s \ + \ \|z_i^p(t_k+T;t_k)\|_{P_i}^2, \quad (3)$$

given constant $b_i \in [0,\infty)$, and matrices $R_i = R_i^T > 0$ and $P_i = P_i^T > 0$. The cost term $b_i\|z_i^p(s;t_k) - \hat{z}_i(s;t_k)\|$ in equation (3) is a state **move suppression term**. It is a way of penalizing the deviation of the predicted state trajectory from the assumed trajectory, which is the trajectory that neighboring agents rely on. In previous works, this term was incorporated into the distributed RHC framework as a constraint, called a consistency constraint [5], [3]. The formulation presented here is an improvement over these past formulations, since the move suppression cost formulation yields an optimization problem that is much easier to solve, and allows a greater degree of freedom to the RHC control law. Note the move suppression term is in the form $b_i\|\cdot\|$, and not $b_i\|\cdot\|^2$. The reason for this is directly related to the form of the cost coupling bound made in part (c) of Assumption 2, which takes the form $c_i\|\cdot\|$ and not $c_i\|\cdot\|^2$. The connection between the move suppression term and the cost coupling bound will be clarified in the stability analysis provided in Section IV.

RHC stability results sometimes rely on the calculation of the optimal cost at each RHC update, e.g., [7], [9]. To relax this requirement here, each cost is minimized while subject to the improvement constraint $J_i(z_i(t_k), u_i^p(\cdot;t_k)) \leq J_i(z_i(t_k), \hat{u}_i(\cdot;t_k))$. The cost $J_i(z_i(t_k), \hat{u}_i(\cdot;t_k))$ is the same as in equation (3), but replacing $(z_i^p(s;t_k), u_i^p(s;t_k))$ with $(\hat{z}_i(s;t_k), \hat{u}_i(s;t_k))$. As will be shown in the coming sections, a feasible solution to this constraint is always available, and the resulting distributed RHC law is stabilizing even without computation of the optimal cost, i.e, feasibility is sufficient for stability. The primary reason to use an improvement constraint for stability, instead of requiring optimality, is to facilitate computationally tractable and feasibility at each RHC update. Other (centralized) RHC methods that also rely on feasibility for stability, instead of optimality, are [1], [10]. The collection of distributed optimal control problems is now defined.

Problem 1: For each agent $i \in \mathcal{V}$ and at any update time t_k, $k \in \mathbb{N}$:

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

Find: a state and control pair $(z_i^p(\tau;t_k), u_i^p(\tau;t_k))$ that minimizes $J_i(z_i(t_k), u_i^p(\cdot;t_k))$ subject to the constraints

$$J_i(z_i(t_k), u_i^p(\cdot;t_k)) \leq J_i(z_i(t_k), \hat{u}_i(\cdot;t_k)) \quad (4)$$

$$\left.\begin{aligned} \dot{z}_i^p(\tau;t_k) &= f_i(z_i^p(\tau;t_k), u_i^p(\tau;t_k)) \\ u_i^p(\tau;t_k) &\in \mathcal{U} \\ z_i^p(\tau;t_k) &\in \mathcal{Z} \end{aligned}\right\} \quad \tau \in [t, t+T],$$

$$z_i^p(t_k+T;t_k) \in \Omega_i(\varepsilon_i) := \{z \in \mathbb{R}^n \mid \|z\|_{P_i}^2 \leq \varepsilon_i\}, \quad (5)$$

with $z_i^p(t_k;t_k) = z_i(t_k)$, and given constant $\varepsilon_i \in (0,\infty)$. ∎

As stated, the constraint (4) is used to guarantee stability. Minimization of the cost J_i is done solely for performance purposes. In practice, the resulting computed control must be feasible, but *need not be optimal*. The closed-loop system for which stability is to be guaranteed is

$$\dot{z}(t) = f(z(t), u_{\text{RH}}(t)), \quad \tau \geq t_0, \quad (6)$$

with the applied *distributed RHC law*

$$u_{\text{RH}}(t) = (u_1^p(t;t_k), ..., u_{N_a}^p(t;t_k)),$$

for $t \in [t_k, t_{k+1})$ and any $k \in \mathbb{N}$. As with most nominally stabilizing RHC formulations [1], [5], [7], [10], observe that, under the closed-loop distributed RHC law, the predicted state $z_i^p(t;t_k)$ for every $i \in \mathcal{V}$ is equal to the actual state $z_i(t)$ for all $t \in [t_k, t_{k+1}]$ and any $k \in \mathbb{N}$. Before stating the control algorithm formally, which in turn defines the assumed trajectories for each agent, decoupled terminal controllers associated with each terminal cost and terminal constraint (5) are required.

Assumption 3: For every agent $i \in \mathcal{V}$, there exists a (possibly nonlinear) state feedback controller $\kappa_i(z_i)$ and a constant $\varepsilon_i \in (0, \infty)$ such that:

(a) $\kappa_i(z_i) \in \mathcal{U}$ for all $z_i \in \Omega_i(\varepsilon_i)$,

(b) $\Omega_i(\varepsilon_i) \subset \mathcal{Z}$, and

(c) the function $V(z) = \sum_{i \in \mathcal{V}} \|z_i\|^2_{P_i}$ satisfies

$$\frac{d}{dt}V(z) \leq -\left[L(z) + \sum_{i \in \mathcal{V}} \|\kappa_i(z_i)\|^2_{R_i}\right],$$

for all $z \in \Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$.

The assumption provides sufficient conditions under which the closed-loop system $\dot{z}_i(t) = f_i(z_i(t), \kappa_i(z_i(t)))$ is asymptotically stabilized to the origin, with constraint feasible state and control trajectories, and positively-invariant region of attraction $\Omega_i(\varepsilon_i)$. Variants of this assumption are common in the RHC literature [9]. For example, by assuming stabilizable and C^2 dynamics f_i for each agent i, a feasible local linear feedback $u_i = K_i z_i$ which stabilizes each linearized and nonlinear subsystem (1) in $\Omega_i(\varepsilon_i)$ can be constructed [1], [10]. Moreover, with this linear feedback control, one can show that ε_i exists for each $i \in \mathcal{V}$ when L is quadratic, in which case the assumption can be converted into an existence lemma. The decoupled feedback controllers $u_i = \kappa_i(z_i)$ are referred to as *terminal controllers*, since they are associated with the terminal state constraint set. With the terminal controllers defined, the assumed trajectories can now also be defined, given by

$$\hat{z}_i(t; t_k) = \begin{cases} z_i^p(t; t_{k-1}), & t \in [t_k, t_{k-1} + T) \\ z_i^\kappa(t; t_{k-1} + T), & t \in [t_{k-1} + T, t_k + T] \end{cases} \tag{7}$$

$$\hat{u}_i(t; t_k) = \begin{cases} u_i^p(t; t_{k-1}), & t \in [t_k, t_{k-1} + T) \\ \kappa_i(z_i^\kappa(t; t_{k-1} + T)), & t \in [t_{k-1} + T, t_k + T] \end{cases} \tag{8}$$

where $z_i^\kappa(\cdot; t_{k-1} + T)$ is the solution to

$$\dot{z}_i^\kappa(t) = f_i(z_i^\kappa(t), \kappa_i(z_i^\kappa(t))), \tag{9}$$

with initial condition $z_i^\kappa(t_{k-1} + T; t_{k-1} + T) = z_i^p(t_{k-1} + T; t_{k-1})$. By construction, each assumed state and control trajectory is the remainder of the previously predicted trajectory, concatenated with the closed-loop response under the terminal controller. Assumption 1(d) is now extended to include the predicted and assumed state trajectories.

Assumption 4: For every agent $i \in \mathcal{V}$ and any update t_k, $k \in \mathbb{N}$, there exists a positive constant $\rho \in (0, \infty)$ such that $\|z_i^p(t; t_k)\| \leq \rho$ and $\|\hat{z}_i(t; t_k)\| \leq \rho$ for all $t \in [t_k, t_k + T]$.

For each $i \in \mathcal{V}$, let $Z_i \subset \mathbb{R}^n$ denote the bounded set of initial states $z_i(t)$ which can be steered to $\Omega_i(\varepsilon_i)$ by a piecewise right-continuous control $u_i^p(\cdot; t) : [t, t + T] \to \mathcal{U}$, with the resulting trajectory $z_i^p(\cdot; t) : [t, t + T] \to \mathcal{Z}$. Note that at initial time, $\hat{z}_i(t; t_0)$ can't be calculated as a function of the previously predicted trajectory, since no such trajectory exists. Therefore, a different means of computing $\hat{z}_i(t; t_0)$ must be defined so that the distributed RHC algorithm can be initialized. The procedure for initialization is incorporated into the control algorithm.

When every agent is in its terminal state constraint set, all agents synchronously switch to their terminal controllers. The terminal controllers are then employed for all future time, resulting in asymptotic stabilization. Switching from RHC to a terminal controller is known as *dual-mode* RHC [10]. To determine if all agents are in their terminal sets, a simple protocol is used. If an agent has reached its terminal set at an update time, it sends a flag message to all other agents. If an agent sends a flag and receives a flag from all other agents, that agents switches to its terminal controller, since this will only happen if all agents have reached their terminal set. The control algorithm is now stated.

Algorithm 1: [Distributed RHC Algorithm] For any agent $i \in \mathcal{V}$, the distributed RHC law is computed as follows:

<u>*Data:*</u> $z_i(t_0) \in Z_i$, $T \in (0, \infty)$, $\delta \in (0, T]$.

<u>*Initialization:*</u> At time t_0, if $z_i(t_0) \in \Omega_i(\varepsilon_i)$, transmit a flag message. If a flag is sent and a flag is received from all other agents, employ the terminal controller κ_i for all future time $t \geq t_0$.

Otherwise, solve a modified Problem 1, setting $b_i = 0$ in (3), $\hat{z}_{-i}(t; t_0) = z_{-i}(t_0)$ for all $t \in [t_0, t_0 + T]$, and removing the constraint (4). Proceed to controller step 1.

<u>*Controller:*</u>

1) Over any interval $[t_k, t_{k+1})$, $k \in \mathbb{N}$:
   a) Apply $u_i^p(\tau; t_k)$, $\tau \in [t_k, t_{k+1})$.
   b) Compute $\hat{z}_i(\cdot; t_{k+1})$ according to (7) and $\hat{u}_i(\cdot; t_{k+1})$ according to (8).
   c) Transmit $\hat{z}_i(\cdot; t_{k+1})$ to every neighbor $j \in \mathcal{N}_i$. Receive $\hat{z}_j(\cdot; t_{k+1})$ from every neighbor $j \in \mathcal{N}_i$, and assemble $\hat{z}_{-i}(\cdot; t_{k+1})$. Proceed to step 2.
2) At any update time t_{k+1}, $k \in \mathbb{N}$:
   a) Measure $z_i(t_{k+1})$.
   b) If $z_i(t_{k+1}) \in \Omega_i(\varepsilon_i)$, transmit a flag to all other agents. If, in addition, a flag is received from all other agents, employ the terminal controller κ_i for all future time $t \geq t_{k+1}$. Otherwise, proceed to step (c).
   c) Solve Problem 1 for $u_i^p(\cdot; t_{k+1})$, and return to step 1. ■

Initialization and part 2(b) of the algorithm presume that the every agent can communicated to all others. To make the algorithm entirely distributed and decentralized, communication of flag messages to neighbors only can be done and a consensus protocol used to determine if all agents are in their terminal sets at any update, as discussed in [2]. At initialization of Algorithm 1, if all agents are not in their terminal sets, a modified Problem 1 is solved in which neighbors are assumed to remain at their initial conditions. While this choice facilitates initialization, it is known that neighbors will not remain at their initial conditions. As such, for performance reasons, it may preferable to reduce the weight of the L_i term in (3). If an iterative procedure can be tolerated, the computed initial $z_i^p(\cdot; t_0)$ and $u_i^p(\cdot; t_0)$ could be defined as the assumed trajectories $\hat{z}_i(\cdot; t_0)$ and $\hat{u}_i(\cdot; t_0)$, with $\hat{z}_i(\cdot; t_0)$ transmitted to neighbors, and the modified Problem 1 resolved again. While the move suppression cost is removed at initialization ($b_i = 0$), the value of b_i is nonzero at every subsequent update t_k, $k \geq 1$, with conditions on permissible values for b_i defined in the next section.

IV. Feasibility and Stability Analysis

This section demonstrates the feasibility and stability properties of the distributed RHC algorithm. For computational reasons, it is desirable for any RHC algorithm to have a feasible solution to the optimal control problem at every update, since the optimization algorithm can then be preempted at each RHC update. A feasible solution at time t_k is defined as a state and control pair $(z_i^p(\cdot; t_k), u_i^p(\cdot; t_k))$ that satisfies all constraints, and results in bounded cost, in Problem 1. According to Algorithm 1, a modified problem is solved at time t_0 (initialization). The remainder of this section assumes that a solution to this problem can be found for every $i \in \mathcal{V}$. Assuming initial feasibility is standard in the RHC literature [1], [9], [10]. The first result of this section is to show that, if a feasible solution to the modified problem is found at t_0, then there is a feasible solution to Problem 1 at every subsequent RHC update time t_k, $k \geq 1$.

Lemma 1: Suppose Assumptions 1–4 hold, and $z_i(t_0) \in Z_i$ for every $i \in \mathcal{V}$. For every agent $i \in \mathcal{V}$, suppose that a feasible solution $(z_i^p(\cdot; t_0), u_i^p(\cdot; t_0))$ to the modified Problem 1 is computed at initial time t_0, with the modified problem defined in the initialization step of Algorithm 1. Then, for every agent $i \in \mathcal{V}$, $(\hat{z}_i(\cdot; t_k), \hat{u}_i(\cdot; t_k))$ is a feasible solution to Problem 1 at every subsequent update time t_k, $k \geq 1$.

The proof is a classical proof, and follows the logic used in [1] and elsewhere. Tthe stability of the closed-loop system (6) is now analyzed.

Theorem 1: Suppose Assumptions 1–4 hold, and $z_i(t_0) \in Z_i$ for every $i \in \mathcal{V}$. Suppose also that a solution to the modified Problem 1 is computed at initial time t_0 for every agent $i \in \mathcal{V}$. Then, by application of Algorithm 1 for all time $t \geq t_0$, the closed-loop system (6) is asymptotically stabilized to the origin, provided the move suppression weight b_i in the cost function (3) satisfies

$$b_i \geq \sum_{j \in \mathcal{N}_i} c_j, \quad \forall i \in \mathcal{V}, \tag{10}$$

where c_i is the strength-of-coupling parameter, defined in Assumption 2.

Equation (10) says that the weight b_i on the move suppression term is bounded from below by the sum of the strength-of-coupling parameters c_j of the neighbors of i. Each strength-of-coupling parameter c_j is a (possibly conservative) measure of how much net coupling there is between j and *all* other neighbors $\mathcal{N}_j$. The larger c_j, and hence the more net coupling, the *more dependent j is on the assumed trajectories of neighbors* in the term L_j in the optimal control cost function[1]. So, another way of interpreting (10) is that, the more the neighbors of i rely on assumed trajectories in their own coupled cost term L_j, the more i must adhere to its own assumed trajectory in the move suppression cost term $b_i\|\cdot\|$. In this way, (10) is a way of ensuring consistency between what neighbors assume an agent does, and what the agent actually does.

[1]While c_j may be large, it may only be a subset of neighbors $\mathcal{N}_j$ in L_j that are dominating the cost, and j would be more dependent on those neighbors assumed trajectories. Still, we can leave this detail aside to interpret (10).

Proof. As with most stability results in RHC theory, a non-negative value function is shown to be strictly decreasing for states outside the terminal constraint sets. Define the value function

$$J(t_k) := \sum_{i \in \mathcal{V}} J_i(z_i(t_k), u_i^p(\cdot; t_k)).$$

By application of Algorithm 1, if $z_i(t_k) \in \Omega_i(\varepsilon_i)$ for all $i \in \mathcal{V}$ at any update time t_k, the terminal controllers take over and stabilize the system to the origin. Therefore, it remains to show that, by application of Algorithm 1, the closed-loop system (6) is driven to the set $\Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$ in finite time.

Suppose the closed-loop system (6) does not enter set $\Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$ in finite time. Then, for any $k \geq 0$, $z_i(t_k) \notin \Omega_i(\varepsilon_i)$ and $z_i(t_{k+1}) \notin \Omega_i(\varepsilon_i)$ for all $i \in \mathcal{V}$. From the cost improvement constraint (4), $J(t_{k+1}) \leq \sum_{i \in \mathcal{V}} J_i(z_i(t_{k+1}), \hat{u}_i(\cdot; t_{k+1}))$, thus, for any $k \in \mathbb{N}$,

$$\begin{aligned}
&J(t_{k+1}) - J(t_k) \;\leq \\
&- \int_{t_k}^{t_{k+1}} \sum_{i \in \mathcal{V}} \left[L_i(z_i^p(s; t_k), \hat{z}_{-i}(s; t_k)) + \|u_i^p(s; t_k))\|_{R_i}^2 \right] \mathrm{d}s \\
&+ \int_{t_{k+1}}^{t_k+T} \sum_{i \in \mathcal{V}} \Big[L_i(\hat{z}_i(s; t_{k+1}), \hat{z}_{-i}(s; t_{k+1})) \\
&\qquad\qquad - L_i(z_i^p(s; t_k), \hat{z}_{-i}(s; t_k)) \Big] \mathrm{d}s \\
&- \int_{t_{k+1}}^{t_k+T} \sum_{i \in \mathcal{V}} b_i \|z_i^p(s; t_k) - \hat{z}_i(s; t_k)\| \; \mathrm{d}s \\
&+ \int_{t_k+T}^{t_{k+1}+T} L(\hat{z}(s; t_{k+1})) + \sum_{i \in \mathcal{V}} \|\kappa_i(\hat{z}_i(s; t_{k+1}))\|_{R_i}^2 \; \mathrm{d}s \\
&+ V(\hat{z}(t_{k+1} + T; t_{k+1})) - V(\hat{z}(t_k + T; t_{k+1})),
\end{aligned}$$

where $L(z) = \sum_{i \in \mathcal{V}} L_i(z_i, z_{-i})$ and $V(z) = \sum_{i \in \mathcal{V}} \|z_i\|_{P_i}^2$. From Assumption 3(c), the collection of last four terms on the right side of the inequality is bounded from above by zero, since $\hat{z}_i(s; t_{k+1}) \in \Omega_i(\varepsilon_i)$ for all $s \in [t_k + T, t_{k+1} + T]$ and every $i \in \mathcal{V}$. Since $\hat{z}_i(s; t_{k+1}) = z_i^p(s; t_k)$ for all $s \in [t_{k+1}, t_k + T]$, from Assumption 2,

$$\begin{aligned}
&\int_{t_{k+1}}^{t_k+T} \sum_{i \in \mathcal{V}} \Big[L_i(\hat{z}_i(s; t_{k+1}), \hat{z}_{-i}(s; t_{k+1})) \\
&\qquad\qquad - L_i(z_i^p(s; t_k), \hat{z}_{-i}(s; t_k)) \Big] \mathrm{d}s \\
&\qquad \leq \int_{t_{k+1}}^{t_k+T} \sum_{i \in \mathcal{V}} c_i \|\hat{z}_{-i}(s; t_{k+1}) - \hat{z}_{-i}(s; t_k)\| \; \mathrm{d}s.
\end{aligned}$$

Therefore, the cost difference satisfies

$$\begin{aligned}&J(t_{k+1}) - J(t_k) + \eta_k \\ &\le \int_{t_{k+1}}^{t_k+T} \sum_{i\in\mathcal{V}} \Big[c_i \|\hat{z}_{-i}(s;t_{k+1}) - \hat{z}_{-i}(s;t_k)\| \\ &\qquad - b_i \|z_i^p(s;t_k) - \hat{z}_i(s;t_k)\|\Big] \mathrm{d}s, \quad (11)\end{aligned}$$

with

$$\eta_k := \sum_{i\in\mathcal{V}} \int_{t_k}^{t_{k+1}} L_i(z_i^p(s;t_k), \hat{z}_{-i}(s;t_k)) + \|u_i^p(s;t_k))\|_{R_i}^2 \ \mathrm{d}s.$$

Next, it is shown that the term on the right of the inequality in (11) is non-positive, provided each b_i satisfies the inequality (10). From the triangle inequality,

$$\begin{aligned}&\sum_{i\in\mathcal{V}} c_i \|\hat{z}_{-i}(s;t_{k+1}) - \hat{z}_{-i}(s;t_k)\| \\ &\quad \le \sum_{i\in\mathcal{V}} c_i \sum_{j\in\mathcal{N}_i} \|\hat{z}_j(s;t_{k+1}) - \hat{z}_j(s;t_k)\| \\ &\quad = \sum_{i\in\mathcal{V}} \Big(\|\hat{z}_i(s;t_{k+1}) - \hat{z}_i(s;t_k)\| \sum_{j\in\mathcal{N}_i} c_j\Big),\end{aligned}$$

and the equality follows since the term $\|\hat{z}_j(s;t_{k+1}) - \hat{z}_j(s;t_k)\|$ is present in the overall summation N_j times, each with one of the coefficients c_l, $l \in \mathcal{N}_j$. If each b_i satisfies (10), the cost difference becomes

$$\begin{aligned}&J(t_{k+1}) - J(t_k) + \eta_k \\ &\le \int_{t_{k+1}}^{t_k+T} \sum_{i\in\mathcal{V}} \Big[\sum_{j\in\mathcal{N}_i} c_j - b_i\Big] \|z_i^p(s;t_k) - \hat{z}_i(s;t_k)\| \mathrm{d}s \le 0.\end{aligned}$$

If $z_i(t_k) \notin \Omega_i(\varepsilon_i)$ and $z_i(t_{k+1}) \notin \Omega_i(\varepsilon_i)$ for all $i \in \mathcal{V}$, for any $k \ge 0$, then $\inf_k \eta_k > 0$. Thus, from the inequality above, if $z_i(t_k) \notin \Omega_i(\varepsilon_i)$ and $z_i(t_{k+1}) \notin \Omega_i(\varepsilon_i)$ for all $i \in \mathcal{V}$, there exists a constant $\eta \in (0, \inf_k \eta_k]$ such that $J(t_{k+1}) \le J(t_k) - \eta$ for any $k \ge 0$. From this inequality, it follows by contradiction that there exists a finite update time t_l such that $z_i(t_l) \in \Omega_i(\varepsilon_i)$ for all $i \in \mathcal{V}$. If this were not the case, the inequality implies $J(t_k) \to -\infty$ as $k \to \infty$. However, $J(t_k) \ge 0$; therefore, there exists a finite time such that the closed-loop system (6) is driven to the set $\Omega_1(\varepsilon_1) \times \cdots \times \Omega_{N_a}(\varepsilon_{N_a})$, concluding the proof. ■

Note that the move suppression term $b_i\|\cdot\|$ in each cost function $J_i(z_i(t_k), u_i^p(\cdot;t_k))$ is not differentiable at the origin. If the optimization algorithm can handle non-smooth costs/constraints, this may not be an issue. To make the move suppression term smooth, one could employ the function $\phi : \mathbb{R}^n \to \mathbb{R}_+$ defined by $\phi(x) = \sqrt{a_1\|x\|^2 + a_2^2} - a_2$, with constants $a_1 > 1$ and $a_2 > 0$ (see [4]).

V. Conclusions

In this paper, a distributed implementation of receding horizon control for coordinated control of multiple agents is formulated. A generic integrated cost function that couples the states of the agents is first defined, where the coupling terms in the cost are required to satisfy a linear growth bound. An example cost function relevant for formation stabilization of multiple autonomous vehicles is shown to satisfy the required bound, provided all state trajectories remain bounded. One aspect of the generality of the approach is that agents dynamics are nonlinear and heterogeneous. The coupling cost is decomposed and distributed optimal control problems are then defined. Each distributed problem is augmented with a *move suppression cost term*, which is a key element in the stability analysis by ensuring that actual and assumed responses of each agent are not too far from one another. Stability of the closed-loop system is proven when the move suppression weight parameter is large enough, in comparison to the linear growth parameters that bound the amount coupling the cost function. The resulting distributed RHC algorithm is computationally scalable, since the size of each local optimal control problem is independent of the number of neighbors for each agent, as well as the total number of agents. The implementation also does not require agents to update sequentially, as in hierarchical methods; agents update their control in parallel. This is likely to be an advantage when the agents are operating in time critical networked environments, in which turn-taking could too taxing to manage.

References

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

[2] W. B. Dunbar. *Distributed Receding Horizon Control of Multiagent Systems*. PhD thesis, California Institute of Technology, Pasadena, CA, 91125, April 2004.

[3] W. B. Dunbar. Distributed receding horizon control of dynamically coupled nonlinear systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, 2007.

[4] W. B. Dunbar. Distributed receding horizon control: Stability via move suppression. In J.S. Shamma, editor, *Cooperative Control of Distributed Multiagent Systems, in press*. John Wiley & Sons, Ltd, 2007.

[5] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006.

[6] E. Franco, T. Parisini, and M. M. Polycarpou. Cooperative control of distributed agents with nonlinear dynamics and delayed information exchange: a stabilizing receding-horizon approach. In *Proc. IEEE Conf. on Decision and Control*, Seville, Spain, 2005.

[7] A. Jadbabaie, J. Yu, and J. Hauser. Unconstrained receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5):776–783, 2001.

[8] T. Keviczky, F. Borrelli, and G. J. Balas. Stability analysis of decentralized rhc for decoupled systems. In *Proc. IEEE Conf. on Decision and Control*, Seville, Spain, 2005.

[9] 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.

[10] H. Michalska and D. Q. Mayne. Robust receeding horizon control of contrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38:1623–1632, 1993.

[11] E. Polak. *Optimization: Algorithms and Consistent Approximations*. Springer, New York, 1997.

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

[13] A. Richards and J. How. Robust model predictive control with imperfect information. In *Proc. American Control Conference*, 2005.

[14] K. Wesselowski and R. Fierro. A dual-mode model predictive controller for robot formations. In *Proc. IEEE Conf. on Decision and Control*, Maui, HI, 2003.