Distributed Receding Horizon Control of Vehicle Platoons: Stability and String Stability

William B. Dunbar, *Member, IEEE*, and Derek S. Caveney, *Member, IEEE*

***Abstract*—This paper considers the problem of distributed control of a platoon of vehicles with nonlinear dynamics. We present distributed receding horizon control algorithms and derive sufficient conditions that guarantee asymptotic stability, leader-follower string stability, and predecessor-follower string stability, following a step speed change in the platoon. Vehicles compute their own control in parallel, and receive communicated position and velocity error trajectories from their immediate predecessor. Leader-follower string stability requires additional communication from the lead car at each update, in the form of a position error trajectory. Predecessor-follower string stability, as we define it, implies leader-follower string stability. Predecessor-follower string stability requires stricter constraints in the local optimal control problems than the leader-follower formulation, but communication from the lead car is required only once at initialization. Provided an initially feasible solution can be found, subsequent feasibility of the algorithms are guaranteed at every update. The theory is generalized for nonlinear decoupled dynamics, and is thus applicable to fleets of planes, robots, or boats, in addition to cars. A simple seven-car simulation examines parametric tradeoffs that affect stability and string stability. Analysis on platoon formation, heterogeneity and size (length) is also considered, resulting in intuitive tradeoffs between lead car and following car control flexibility.**

***Index Terms*—Distributed control, intelligent vehicles, path planning.**

I. Introduction

We are interested in the control of a platoon of dynamically decoupled vehicle systems that are required to stably and cooperatively perform group speed changes. In particular, the closed-loop response must be string stable to ensure attenuation of the closed-loop position error, from the first to the last vehicle in the platoon. We explore receding horizon control [15], in which the current control action is determined by solving online, at each sampling instant, a finite-horizon optimal-control problem. In continuous-time formulations, each optimization yields an open-loop control trajectory and the initial portion of the trajectory is applied to the system until the next receding horizon update time. For the problem of interest here, cooperation between subsystems can be incorporated in the optimal control problem by including coupling terms in the cost function or constraints, as done in [7] and [11]. In this paper, a *distributed* receding horizon controller is presented in which each vehicle is assigned its own optimal control problem, optimizes only for its own control during each update period, and transmits and receives information only with the vehicle behind and ahead, respectively. For the purpose of establishing string stability, communication with the platoon's lead car is also required, but only in the form of position and speed error trajectories, i.e., *no information about the lead car's acceleration or control profile is required for string stability*. This is one way in which our controller is different than many of the existing string-stabilizing controllers.

The automotive industry is embracing receding horizon control research for powertrain [19] and vehicle stability [8] applications. Furthermore, this research is being applied to path planning applications for autonomous driving [20] and eco-driving [12]. However, to the authors' knowledge, this work is the first to apply distributed receding horizon control to the automotive domain. The convergence of advanced global positioning technologies, prototype vehicle-to-vehicle communication, and increased on-board computation capabilities has allowed academia and industry to explore the possibilities of cooperative control between cars. In principle, distributed receding horizon control enables cooperative control by providing implicit robustness to communication dropouts, enforcing constraints on allowable vehicle motion, and sharing predicted paths between vehicles. Global positioning further enables cooperative control through the existence of a common global clock and inertial coordinate frame.

Literature on distributed receding horizon control has examined coupled subsystem dynamics for linear dynamics [1], [3], [17], [23] and nonlinear dynamics [6], [14]. Distributed receding horizon control of multiple decoupled vehicles has also been explored for linear [13], [18] and nonlinear [7], [9], [11] vehicle dynamics, with coupling in cost functions and/or constraints. To our knowledge, this paper is the first to address string stability of vehicle platoons with nonlinear dynamics using distributed receding horizon control. We do not survey the extensive literature on string stability here, much of which is applicable to linear dynamics, but point interested readers to the seminal paper by Swaroop and Hedrick that is applicable to a countably infinite number of vehicles with nonlinear dynamics [22]. By comparison with other string stable controllers, receding horizon control has the often cited advantage that a variety of constraints on the states and inputs can in principle be accommodated. We note that there are other methods such as controlled invariance that can handle saturation in the inputs

Manuscript received July 16, 2010; revised December 10, 2010, April 08, 2011, and April 15, 2011; accepted June 06, 2011. Date of publication June 16, 2011; date of current version February 29, 2012. The work of W. B. Dunbar was supported by Toyota Motor Engineering and Manufacturing North America as a consultant. Recommended by Associate Editor A. Ferrara.

W. B. Dunbar is with the Department of Computer Engineering, University of California, Santa Cruz, CA 95064 USA (e-mail: dunbar@soe.ucsc.edu).

D. S. Caveney is with the Toyota Motor Engineering and Manufacturing North America, Inc., Ann Arbor, MI 48105 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2011.2159651

while addressing platoon stability for spatially invariant systems [21], such as vehicle platoon with identical linear vehicle dynamics.

The platoon stabilizing controller examined here is an improvement over [7]. In [7], controller updates were assumed to happen instantaneously and synchronously, whereas the method here allows dedicated times for communication and computation during each update period, with permission of local steps to be asynchronous provided the time stamps for the start of update periods to be globally common (which can be facilitated by a GPS-based clock, for example). As in [6] and [7], vehicles generate and transmit an assumed trajectory that is based on the extension of the previously computed trajectory. Also similar to [6] and [7], cost and constraint penalties on the deviation between computed and assumed trajectories are used in the local optimal control problems to ensure stability, and for the first time in this work, string stability. Such penalties act as a "contract" of sorts, between neighbors, that vehicles are willing to self-impose to ensure boundedness and eventually convegence to zero of the deviation between implemented and assumed behaviors, a mechanism that is woven into how neighbors are modeled in [21].

We begin in Section II by defining the nonlinear subsystem dynamics, and the distributed optimal control problems are presented in Section III. The distributed receding horizon control algorithms are then defined, and the stability and string stability results are given in Section IV. Finally, Section VIII discusses conclusions and extensions.

II. System Description and Objective

In this section, we define the system dynamics and pose an integrated cost function for multivehicle platoon stabilization. The states of the vehicles are coupled in the cost function, while each vehicle has decoupled dynamics. We make use of the following notation. The symbol $\|Qx\|_p$ will denote the $p=1$ or ∞ vector norm of the vector $Qx \in \mathbb{R}^n$, or $(x'Qx)^{1/2}$ when $p=2$, with Q as any symmetric and positive definite or semi-definite real matrix.

Our objective is to stabilize a platoon of vehicles $i \in \{1,\ldots,N_v\}$, with lead vehicle $i=1$, and each vehicle $j \geq 2$ having predecessor $j-1$. For each vehicle i, the position and speed are denoted by $(q_i(t), \dot{q}_i(t)) \in \mathbb{R}^2$. The lead car will track a constant speed reference trajectory $(q_L(t), \dot{q}_L)$, with $q_L(t) = q_L(0) + t\dot{q}_L$ for $t \geq 0$. The reference trajectory for each following car $j \geq 2$ is $(q_L(t) - (j-1)d_{\text{sep}}, \dot{q}_L)$, with d_{sep} equal to the constant desired separation distance between vehicles. With this notation, we can define the error states

$$z_i(t) = \begin{bmatrix} q_{i,\mathrm{e}}(t) \\ \dot{q}_{i,\mathrm{e}}(t) \end{bmatrix} = \begin{bmatrix} q_i(t) - q_L(t) + (i-1)d_{\text{sep}} \\ \dot{q}_i(t) - \dot{q}_L \end{bmatrix} \tag{1}$$

and control input $u_i(t) \in \mathbb{R}$. The decoupled second-order, time-invariant nonlinear error dynamics for each vehicle $i \in \{1,\ldots,N_v\}$ are denoted

$$\dot{z}_i(t) = f_i\left(z_i(t), u_i(t)\right), \quad t \geq 0. \tag{2}$$

In this paper, we assume there are no model errors or external disturbances. We now make some standard assumptions regarding the system (2).

Assumption II.1: The following holds for all i: a) $f_i : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ is twice continuously differentiable, $0 = f_i(0,0)$, and f_i linearized around $(0,0)$ is controllable; b) the system (2) has a unique, absolutely continuous solution for any initial condition $z_i(0)$ and any piecewise right-continuous control $u : [0,\infty) \to \mathbb{R}$.

We now define the stability and string stability control problems.

Definition 1 (Stability Control Problem): For a step change in for open faced character $\dot{q}_L$ at time $t=0$, the control objective is to asymptotically stabilize the error state of every car in the platoon to the origin.

Observe that a step change in $\dot{q}_L$ at $t=0$ implies that $z_i(0) \neq 0, \forall i = 1,\ldots,N_v$.

Definition 2 (Leader-Follower String Stability Control Problem: For a step change in $\dot{q}_L$ at time $t=0$, the control objective is to asymptotically stabilize the state of every car in the platoon to the origin, and satisfy the following condition: For every $j = 2,\ldots,N_v$, there exists a constant $\alpha_j \in (0,1)$ such that the closed-loop position error satisfies

$$\max_{t\geq 0} |q_{j,\mathrm{e}}(t)| \leq \alpha_j \max_{t\geq 0} |q_{1,\mathrm{e}}(t)|. \tag{3}$$

A more-stringent condition that will also be examined is predecessor-follower string stability.

Definition 3 (Predecessor-Follower String Stability Control Problem): For a step change in $\dot{q}_L$ at time $t=0$, the control objective is to asymptotically stabilize the state of every car in the platoon to the origin, and satisfy the following condition: For every $j = 2,\ldots,N_v$, there exists a constant $\alpha_j \in (0,1)$ such that the closed-loop position error satisfies

$$\max_{t\geq 0} |q_{j,\mathrm{e}}(t)| \leq \alpha_j \max_{t\geq 0} \left|q_{(j-1),\mathrm{e}}(t)\right|. \tag{4}$$

By Definition 2, leader-follower string stability permits any follower to have a larger closed-loop position error than any other car, including it's immediate predecessor, *except* the lead car. Thus, it is less stringent than the more classical predecessor-follower string stability condition in Definition 3. As we have defined it, predecessor-follower string stability implies leader-follower string stability.

III. Distributed Receding Horizon Control

In this section, we introduce notation and define the optimal control problem and the distributed receding horizon control algorithm for each vehicle. 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 the sample interval. The common receding horizon update times are denoted $t_k = t_0 + \delta k$, where $t_0 = 0$ and $k \in \mathbb{N} = \{0,1,2,\ldots\}$.

We have that $z_i(t)$ and $u_i(t)$ are the actual error state and control, respectively, for each vehicle $i \in \{1,\ldots,N_v\}$ at any time $t \geq t_0$. Over any prediction interval $[t_k, t_k + T]$, $k \in \mathbb{N}$, associated with update time t_k, we denote three trajectories:

$z_i^p(\tau; t_k)$: the predicted state trajectory
$z_i^*(\tau; t_k)$: the optimal predicted state trajectory
$\hat{z}_i(\tau; t_k)$: the assumed state trajectory

where $\tau \in [t_k, t_k + T]$. The corresponding control trajectories are denoted likewise. We use $z_i^p(\tau; t_k)$ as the state trajectory variable that parameterizes the optimal control problem, and $z_i^*(\tau; t_k)$ as the optimal solution to that problem. We use $\hat{z}_i(\tau; t_k)$ as the trajectory communicated to follower $i+1$, and which $i+1$ *assumes* that i will implement over the update period $[t_k, t_{k+1}]$. As detailed below, at time $\tau = t_k$, all of these trajectories are equal to the initial condition $z_i(t_k)$. Additionally, a terminal equality constraint is implemented in each optimal control problem, and at time $\tau = t_k + T$, all of these trajectories are equal to zero. The closed-loop system for which stability is to be guaranteed is

$$\dot{z}_i^*(t; t_k) = f_i\left(z_i^*(t; t_k), u_i^*(t; t_k)\right), \quad t \in [t_k, t_{k+1}],\ k \in \mathbb{N},\ i \in \{1, \ldots, N_v\}. \tag{5}$$

We now define the distributed optimal control problems.

Problem III.1: For each vehicle $i \in \{1, \ldots, N_v\}$ and update time t_k, $k \in \mathbb{N}$, given $z_i(t_k)$, $\hat{z}_i(\tau; t_k)$, and $\hat{z}_{i-1}(\tau; t_k)$ for all $\tau \in [t_k, t_k + T]$, find

$$J_i^*\left(z_i(t_k), u_i^*(\cdot; t_k)\right) = \min_{u_i^p(\cdot; t_k)} J_i\left(z_i^p(\cdot; t_k), u_i^p(\cdot; t_k)\right)$$

such that

$$\begin{aligned} \dot{z}_i^p(\tau; t_k) &= f_i\left(z_i^p(\tau; t_k), u_i^p(\tau; t_k)\right), \quad \tau \in [t_k, t_k + T] \\ z_i^p(t_k; t_k) &= z_i(t_k), \quad z_i^p(t_k + T; t_k) = 0 \end{aligned}$$

where

$$\begin{aligned} J_i\left(z_i^p(\cdot; t_k), u_i^p(\cdot; t_k)\right) = \int_{t_k}^{t_k+T} \Big\{ &\left\|Q_i z_i^p(s; t_k)\right\|_p \\ &+ \left\|F_i\left(z_i^p(s; t_k) - \hat{z}_i(s; t_k)\right)\right\|_p \\ &+ \left\|G_i\left(z_i^p(s; t_k) - \hat{z}_{i-1}(s; t_k)\right)\right\|_p \\ &+ \left\|R_i u_i^p(s; t_k)\right\|_p \Big\}\, \mathrm{d}s. \end{aligned}$$

All weighting matrices are assumed symmetric and satisfy the following:

- $Q_1 > 0$, $F_1 \geq 0$, $G_1 = 0$, and $R_1 > 0$ (for the lead car);
- $Q_i \geq 0$, $F_i \geq 0$, $G_i \geq 0$, and $R_i > 0$, for $i = 2, \ldots, N_v$.

We refer to the (optimal) cost terms as the *move suppression (MS) term* $\|F_i(z_i^* - \hat{z}_i)\|$ and the *predecessor relative error (PRE) term* $\|G_i(z_i^* - \hat{z}_{i-1})\|$. After substitution, the difference states in these terms have the following form:

$$z_i^* - \hat{z}_i = \begin{bmatrix} q_i^* - \hat{q}_i \\ \dot{q}_i^* - \dot{\hat{q}}_i \end{bmatrix}, \quad z_i^* - \hat{z}_{i-1} = \begin{bmatrix} q_i^* - \hat{q}_{i-1} + d_{\text{sep}} \\ \dot{q}_i^* - \dot{\hat{q}}_{i-1} \end{bmatrix}. \tag{6}$$

Remark III.1: If $Q_i = 0$ for any following car $i \geq 2$, the optimal control problem could be rewritten to include *only information from the preceding car* [see (6)], with no information about the reference $(q_L(t), \dot{q}_L(t))$ required. Removing the requirement that all cars in the platoon have knowledge about the lead car's reference trajectory would have obvious advantages in real applications. To achieve this, the cost function, initial condition and dynamics in optimal control problem would be rewritten in the inertial states $(q_i^*, \dot{q}_i^*)$, instead of the error states. Next, the terminal constraint could equivalently be defined using the predecessor's trajectories as $q_i^*(t_k + T; t_k) = \hat{q}_{i-1}(t_k + T; t_k) - d_{\text{sep}}$ and $\dot{q}_i^*(t_k + T; t_k) = \dot{\hat{q}}_{i-1}(t_k + T; t_k)$, provided that $i-1$ has already been informed of any changes in desired speed through their predecessor's assumed trajectories. For this reason, we will pursue a stability analysis that holds while permitting $Q_i = 0$ for any following car $i \geq 2$. We also note that the Q_i terms could be re-written using the computed lead-car trajectories, instead of the (perfect) reference trajectories.

The terminal equality constraint $z_i^p(t_k + T; t_k) = 0$ facilitates the stability analysis. Instead of a terminal equality constraint, much research has examined receding horizon controller stability while implementing a terminal inequality constraint, in which case $z_i^p(t_k + T; t_k)$ is constrained to arrive in a compact set containing the origin [4], [16]. Principally, the motivation is that such a constraint would be easier to enforce numerically in the optimization solver than an equality constraint. There are efficient modern methods of solving nonlinear optimal control problems with terminal equality constraints [2], [20], and so we do not view our use of a terminal equality constraint as a serious drawback. We now state the control algorithm that all cars implement in parallel during every (common) update period $[t_k, t_{k+1}]$.

Algorithm III.1: Distributed receding horizon controller for any vehicle $i \in \{1, \ldots, N_v\}$:

Data: $T \in (0, \infty)$, $\delta \in (0, T]$. Platoon is at equilibrium ($z_i = 0$, $\forall i$) for all time $t < 0$.

Initialization: At some time $t' < 0 - \delta$, the lead car $i = 1$ broadcasts the intent to change desired speed to a new constant value, starting at time $t_0 = 0$. All cars receive the new desired speed $\dot{q}_L$. During the initialization update period $[-\delta, 0]$, each car i solves Problem III.1 setting $F_i = G_i = 0$ and using the new desired speed.

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

1) Apply $u_i^*(\tau; t_k)$, $\tau \in [t_k, t_{k+1})$.
2) Compute predicted state $z_i(t_{k+1}) = z_i^*(t_{k+1}; t_k)$.
3) Compute $\hat{z}_i(\tau; t_{k+1})$ as

$$\hat{z}_i(\tau; t_{k+1}) = \begin{cases} z_i^*(\tau; t_k), & \tau \in [t_{k+1}, t_k + T) \\ 0, & \tau \in [t_k + T, t_{k+1} + T]. \end{cases} \tag{7}$$

4) Transmit $\hat{z}_i(\cdot; t_{k+1})$ to follower $i+1$ and for $j \geq 2$ receive $\hat{z}_j(\cdot; t_{k+1})$ from predecessor $j = i - 1$.
5) Solve Problem III.1 for $u_i^*(\cdot; t_{k+1})$.

An assumed control $\hat{u}_i(\tau; t_{k+1})$ is defined likewise and will be used in the analysis sections

$$\hat{u}_i(\tau; t_{k+1}) = \begin{cases} u_i^*(\tau; t_k), & \tau \in [t_{k+1}, t_k + T) \\ 0, & \tau \in [t_k + T, t_{k+1} + T]. \end{cases} \tag{8}$$

During the initialization update period $[-\delta, 0]$, each car i solves Problem III.1 setting $F_i = G_i = 0$ and using the new desired speed. As such, the assumed trajectories are not needed during the initialization phase. There are several other forms of initialization that could be considered. For example, the lead car could solve Problem III.1 setting $F_1 = G_1 = 0$, then broadcast $z_i^*(\cdot; t_0)$ to all cars in the platoon. Subsequently, all cars $i \geq 2$ could solve Problem III.1 removing the move suppression term ($F_i = 0$) and redefining the predecessor-relative error term as a lead-relative error term, and set $G_i > 0$. This two-stage optimization at initialization (one optimization for the leader, and then all followers optimizing in parallel) would need to terminate and converge before $t_0 = 0$.

IV. Analysis

A. Stability

We denote the concatenated vectors $z = (z_1, \ldots, z_{N_v})$ and $u = (u_1, \ldots, u_{N_v})$, using comparable notation for the optimal and assumed trajectories. The main result of this subsection is to show that by applying Algorithm III.1, the closed-loop state $z(t)$ converges to the origin asymptotically. At any time t_k, $k \in \mathbb{N}$, the sum of the optimal distributed value functions is denoted

$$J_\Sigma^*(z(t_k)) = \sum_{i=1}^{N_v} J_i^*\left(z_i(t_k), u_i^*(\cdot; t_k)\right).$$

The following is a standard result from receding horizon control theory.

Lemma IV.1: Suppose Assumption II.1 holds. Then, for any update period $\delta \in (0, T]$, Problem III.1 has

$$\left(z_i^p(\cdot; t_k), u_i^p(\cdot; t_k)\right) = \left(\hat{z}_i(\cdot; t_k), \hat{u}_i(\cdot; t_k)\right)$$

as a feasible solution at any update time t_k, $k \in \{1, 2, \ldots\}$.

The assumed control $\hat{u}_i(\cdot; t_k)$ defined in (8) is exactly the feasible control trajectory used in [4] and elsewhere. The remaining analysis is based on Algorithm III.1, and so relies on computing the optimal solution to Problem III.1 at every update. The following lemma gives a bounding result on the decrease in $J_\Sigma^*(\cdot)$ from one update to the next.

Lemma IV.2: Suppose Assumption II.1 holds, and let $p = 1$, 2 or ∞ in the norms of the local cost functions. If the MS and PRE weights satisfy the bounds

$$F_i \geq G_{i+1}, \qquad i = 1, \ldots, N_v - 1 \tag{9}$$

i.e., if the PRE weight of each car $i+1$ is upper bounded by the MS weight of its predecessor i, then by application of Algorithm III.1, the function $J_\Sigma^*(\cdot)$ satisfies

$$J_\Sigma^*(z(t_k + \delta)) - J_\Sigma^*(z(t_k)) \leq -\int_{t_k}^{t_k+\delta} L\left(z^*(s; t_k), \hat{z}(s; t_k), u^*(s; t_k)\right) ds$$

for all $k \in \{1, 2, \ldots\}$, where

$$L(z^*, \hat{z}, u^*) \triangleq \sum_{i=1}^{N_v} \|Q_i z_i^*\|_p + \|F_i(z_i^* - \hat{z}_i)\|_p + \|G_i(z_i^* - \hat{z}_{i-1})\|_p + \|R_i u_i^*\|_p.$$

Proof: For any $k \geq 1$

$$J_\Sigma^*(z(t_k)) = \int_{t_k}^{t_k+T} L\left(z^*(s; t_k), \hat{z}(s; t_k), u^*(s; t_k)\right) \mathrm{d}s.$$

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

$$\begin{aligned} J_\Sigma^*(z(t_{k+1})) &\leq \int_{t_{k+1}}^{t_{k+1}+T} L\left(\hat{z}(s; t_{k+1}), \hat{z}(s; t_{k+1}), \hat{u}(s; t_{k+1})\right) \mathrm{d}s \\ &= \int_{t_{k+1}}^{t_k+T} L\left(z^*(s; t_k), z^*(s; t_k), u^*(s; t_k)\right) \mathrm{d}s. \end{aligned}$$

The equality holds because of how $\hat{z}(s; t_{k+1})$ and $\hat{u}(s; t_{k+1})$ are defined by (7) and (8), respectively. Subtracting the equations gives

$$\begin{aligned} &J_\Sigma^*(z(t_k + \delta)) - J_\Sigma^*(z(t_k)) \\ &\quad \leq -\int_{t_k}^{t_{k+1}} L\left(z^*(s; t_k), \hat{z}(s; t_k), u^*(s; t_k)\right) \mathrm{d}s \\ &\quad + \int_{t_{k+1}}^{t_k+T} L\left(z^*(s; t_k), z^*(s; t_k), u^*(s; t_k)\right) \\ &\quad - L\left(z^*(s; t_k), \hat{z}(s; t_k), u^*(s; t_k)\right) \mathrm{d}s. \end{aligned}$$

Showing that the second integral expression with limits t_{k+1} to $t_k + T$ is bounded above by 0 gives the stated result. Combining terms from the definition of L we have that the second term's integrand is equal to

$$\begin{aligned} \sum_{i=1}^{N_v} &\left\|G_i\left(z_i^*(s; t_k) - z_{i-1}^*(s; t_k)\right)\right\|_p \\ &- \left\|F_i\left(z_i^*(s; t_k) - \hat{z}_i(s; t_k)\right)\right\|_p \\ &- \left\|G_i\left(z_i^*(s; t_k) - \hat{z}_{i-1}(s; t_k)\right)\right\|_p. \end{aligned} \tag{10}$$

With the triangle inequality for vector norms

$$\begin{aligned} \left\|G_i\left(z_i^*(s; t_k) - z_{i-1}^*(s; t_k)\right)\right\|_p &\leq \left\|G_i\left(z_i^*(s; t_k) - \hat{z}_{i-1}(s; t_k)\right)\right\|_p \\ &+ \left\|G_i\left(z_{i-1}^*(s; t_k) - \hat{z}_{i-1}(s; t_k)\right)\right\|_p. \end{aligned}$$

With $G_1 = 0$, we have (10) is bounded by

$$\sum_{i=1}^{N_v-1} \|G_{i+1}(z_i^*(s;t_k) - \hat{z}_i(s;t_k))\|_p - \|F_i(z_i^*(s;t_k) - \hat{z}_i(s;t_k))\|_p$$

and this is bounded above by zero if (9) holds. This completes the proof. ■

Equation (9) provides an interesting tradeoff between the PRE and MS terms in neighboring local cost functions. Essentially, the sufficient condition for stability suggests that a following car can commit more to the PRE term (i.e., can invest more in the assumed behavior of its predecessor) only if the increased weight is matched or exceeded by the commitment of its predecessor to its own assumed trajectory via the MS term. Another viewpoint: car i sends an assumed trajectory to its follower; with little or no commitment to that trajectory (small F_i), the follower $i+1$ should not rely too heavily on the received trajectory (G_{i+1} must be as small or smaller than F_i) to be consistent with the stability criteria. It is important to emphasize that Lemma IV.2 is distributed with respect to the vehicles in the platoon. Vehicle i only requires knowledge of its immediate predecessor's weight F_{i-1} to restrict its choice of G_i.

Typically, the norm terms in the integrated cost function are 2-norms squared. The proof of Lemma IV.2 must be modified for that case, since use of the triangle inequality following (10) results in cross terms that must be compensated for. A proof that achieves this can be done by following the approach in [7]. The first main theorem of the paper is now stated.

Theorem 1: Suppose Assumption II.1 and the conditions of Lemma IV.2 hold. Then, by application of Algorithm III.1, the closed-loop state trajectory enters the compact level set

$$\Omega_\beta = \left\{ z \in \mathbb{R}^{2N_v} | J_\Sigma^*(z) \leq \beta \right\}$$

in finite time, for any given $\beta > 0$, and remains there for all future time. Additionally, if $u^*(\cdot; z)$ is continuous in z at $z = 0$, then the closed-loop state trajectories are asymptotically stabilized to the origin.

Proof: Let $V(z) = J_\Sigma^*(z)$ serve as a Lyapunov certificate. The bound in Lemma IV.2 shows $V(z(t_k))$ decreasing strictly monotonically with update time t_k. This can be used to prove finite-time convergence to and positive invariance of Ω_β, following arguments established in [4], [7]. The function $V(z)$ satisfies $V(0) = 0$ and $V(z) > 0$ for $z \neq 0$. With the assumption on continuity of the control at the origin, one can show $V(z)$ is continuous at $z = 0$ [5]. From these properties, asymptotic stabilization follows using V as the Lyapunov certificate [4]. ■

B. Leader-Follower String Stability

In the analysis here, we examine two different methods for achieving leader-follower string stability. In each method, we redefine the local optimal control problems and algorithm presented above. The first method places conditions on new design parameters, and requires an assumption about the maximum closed-loop position error of the leader. The second method places more restrictive conditions on the same design parameters, and requires a harder optimization problem at initialization, but removes the need for the assumption on the maximum closed-loop position error of the leader. In this section, we use

$$\left|q_{j,\mathrm{e}}^*(t;t_k)\right|_\infty = \max_{t\in[t_k,t_k+T]} \left|q_{j,\mathrm{e}}^*(t;t_k)\right|$$
$$\left|q_{j,\mathrm{e}}^*(t;t_k)\right|_{\infty,\delta} = \max_{t\in[t_k,t_k+\delta]} \left|q_{j,\mathrm{e}}^*(t;t_k)\right|.$$

For any car $j = 1, \ldots, N_v$, the maximum closed-loop position error satisfies $\max_{t\geq 0} |q_{j,\mathrm{e}}(t)| = \max_{l\in\{0,1,2\ldots\}} |q_{j,\mathrm{e}}^*(t;t_l)|_{\infty,\delta}$ by definition, and since $|q_{j,\mathrm{e}}^*(t;t_k)|_{\infty,\delta} \leqq |q_{j,\mathrm{e}}^*(t;t_k)|_\infty$ we also have that $\max_{t\geq 0} |q_{j,\mathrm{e}}(t)| \leq \max_{l\in\{0,1,2\ldots\}} |q_{j,\mathrm{e}}^*(t;t_l)|_\infty$. If the maximum computed error occurred during an update period, then we have equality: $\max_{t\geq 0} |q_{j,\mathrm{e}}(t)| = \max_{l\in\{0,1,2\ldots\}} |q_{j,\mathrm{e}}^*(t;t_l)|_\infty$. Assuming this for the lead car facilitates the algorithm and analysis, as shown in Method 1.

1) Method 1: For the method developed, we show that for any $j = 2, \ldots, N_v$ there exists an $\alpha_j \in (0,1)$ such that the closed-loop position error satisfies

$$\left|q_{j,\mathrm{e}}^*(t;t_k)\right|_\infty \leq \alpha_j \max_{l\in\{0,\ldots,k\}} \left|q_{1,\mathrm{e}}^*(t;t_l)\right|_\infty, \quad \forall k = 0,1,2,\ldots \tag{11}$$

Equation (11) is a sufficient condition that ensures (3) with the following assumption:

Assumption IV.1: The maximum computed position error for the lead car occurs during an update period.

As stated, Assumption IV.1 implies $\max_{t\geq 0} |q_{j,\mathrm{e}}(t)| = \max_{l\in\{0,1,2\ldots\}} |q_{j,\mathrm{e}}^*(t;t_l)|_\infty$. An alternate algorithm that does not require this assumption is presented in the next section on Method 2; however, that algorithm incorporates more constraints to be enforced at initialization.

Algorithm IV.1: Distributed receding horizon controller for any vehicle $i \in \{1, \ldots, N_v\}$:

Data: $T \in (0, \infty)$, $\delta \in (0, T]$. Platoon is at equilibrium ($z_i = 0$, $\forall i$) for all time $t < 0$.

Initialization with update period $[-\delta, 0]$: At some time $t' < 0 - \delta$, the lead car $i = 1$ broadcasts the intent to change desired speed to a new constant value, starting at time $t_0 = 0$. All cars receive the new desired speed $\dot{q}_L$. In addition, during the initialization update period $[-\delta, 0]$:

1) Car 1 solves Problem III.1 setting $F_1 = 0$ and using the new desired speed, then broadcasts $q_{1,\mathrm{e}}^*(\cdot;t_0)$ to all cars $2, \ldots, N_v$ in the platoon. Next:
2) Each car $i \geq 2$ solves Problem III.1 with the additional constraint:

$$\left|q_{i,\mathrm{e}}^p(t;t_0)\right|_\infty \leq \beta_i \left|q_{1,\mathrm{e}}^*(t;t_0)\right|_\infty \tag{12}$$

setting $\beta_i \in (0,1)$, $F_i = G_i = 0$ and using the new desired speed.

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

1) Apply $u_i^*(\tau; t_k)$, $\tau \in [t_k, t_{k+1})$.
2) Compute predicted state $z_i(t_{k+1}) = z_i^*(t_{k+1}; t_k)$.
3) Compute $\hat{z}_i(\tau; t_{k+1})$ as defined by (7).
4) Transmit $\hat{z}_i(\cdot; t_{k+1})$ to follower $i+1$ and for $j \geq 2$ receive $\hat{z}_j(\cdot; t_{k+1})$ from predecessor $j = i - 1$. Additionally, all $i \geq 2$ receive $\hat{q}_{1,\mathrm{e}}(\cdot; t_{k+1})$ from the lead car.

5) Solve Problem III.1 for $u_i^*(\cdot;t_{k+1})$, with the additional constraint:

$$\left|q_{i,\mathrm{e}}^p(t;t_{k+1}) - \hat{q}_{i,\mathrm{e}}(t;t_{k+1})\right|_\infty \le \varepsilon_{i,k+1}\left|\hat{q}_{1,\mathrm{e}}(t;t_{k+1})\right|_\infty \tag{13}$$

setting the update-dependent parameter $\varepsilon_{i,k+1} > 0$.

Theorem 2: Suppose Assumption II.1 holds, the conditions of Lemma IV.2 holds, and that the application of Algorithm II.1 gives a feasible solution $(z_i^*(\cdot;t_0), u_i^*(\cdot;t_0))$ at initialization for every $i = 1,\dots,N_v$. Then, by application of Algorithm II.1:

1) The assumed trajectories $(z_i^p(\cdot;t_k), u_i^p(\cdot;t_k)) = (\hat{z}_i(\cdot;t_k), \hat{u}_i(\cdot;t_k))$ are a feasible solution at every subsequent update $k \ge 1$, and for every $i = 1,\dots,N_v$;
2) The stability properties of Algorithm III.1 via Theorem 1 are preserved.
3) The closed-loop position errors satisfy (11) if the following parametric condition holds:

$$\beta_i + \sum_{k=1}^{\infty} \varepsilon_{i,k}(1+\varepsilon_{1,k}) < 1, \qquad \forall i = 2,\dots,N_v. \tag{14}$$

Proof: Part 1 follows from Lemma IV.1 and that the assumed trajectories trivially satisfy (13) for all $i = 1,\dots,N_v$. For part 2, the logic in Lemma IV.2 and Theorem 1 still applies since the cost function remains unchanged for all local optimal control problems. We now demonstrate part 3. From (12) at initialization $(k = 0)$

$$\left|q_{i,\mathrm{e}}^*(t;t_0)\right|_\infty \le \beta_i \left|q_{1,\mathrm{e}}^*(t;t_0)\right|_\infty . \tag{15}$$

From the triangle inequality and (13) and using $|q_{i,\mathrm{e}}^*(t;t_{k-1})|_\infty \ge |\hat{q}_{i,\mathrm{e}}(t;t_k)|_\infty$, for any $i = 2,\dots,N_v$ and any $k \ge 1$, we have

$$\begin{aligned}\left|q_{i,\mathrm{e}}^*(t;t_k)\right|_\infty &\le \left|q_{i,\mathrm{e}}^*(t;t_k) - \hat{q}_{i,\mathrm{e}}(t;t_k)\right|_\infty + \left|\hat{q}_{i,\mathrm{e}}(t;t_k)\right|_\infty \\ &\le \varepsilon_{i,k}\left|\hat{q}_{1,\mathrm{e}}(t;t_k)\right|_\infty + \left|q_{i,\mathrm{e}}^*(t;t_{k-1})\right|_\infty .\end{aligned} \tag{16}$$

From the triangle inequality and (13) for lead car $i = 1$

$$\left|\hat{q}_{1,\mathrm{e}}(t;t_k)\right|_\infty \le \varepsilon_{1,k}\left|q_{1,\mathrm{e}}^*(t;t_{k-1})\right|_\infty + \left|q_{1,\mathrm{e}}^*(t;t_k)\right|_\infty .$$

Combining this with (16) gives

$$\begin{aligned}\left|q_{i,\mathrm{e}}^*(t;t_k)\right|_\infty \le \varepsilon_{i,k}(1+\varepsilon_{1,k}) \max_{l=k-1,k}\left|q_{1,\mathrm{e}}^*(t;t_l)\right|_\infty \\ + \left|q_{i,\mathrm{e}}^*(t;t_{k-1})\right|_\infty .\end{aligned} \tag{17}$$

With (15), (17) for $k = 1$ can be expressed as

$$\left|q_{i,\mathrm{e}}^*(t;t_1)\right|_\infty \le (\varepsilon_{i,1}(1+\varepsilon_{1,1}) + \beta_i) \max_{l=0,1}\left|q_{1,\mathrm{e}}^*(t;t_l)\right|_\infty . \tag{18}$$

With (18), (17) for $k = 2$ can be expressed as

$$\begin{aligned}\left|q_{i,\mathrm{e}}^*(t;t_2)\right|_\infty \le (\varepsilon_{i,1}(1+\varepsilon_{1,1}) + \varepsilon_{i,2}(1+\varepsilon_{1,2}) + \beta_i) \\ \times \max_{l=0,1,2}\left|q_{1,e}^*(t;t_l)\right|_\infty .\end{aligned} \tag{19}$$

Carrying this out recursively, we have

$$\left|q_{i,\mathrm{e}}^*(t;t_n)\right|_\infty \le \left(\beta_i + \sum_{k=1}^{n} \varepsilon_{i,k}(1+\varepsilon_{1,k})\right) \max_{l=0,\dots,n}\left|q_{1,e}^*(t;t_l)\right|_\infty . \tag{20}$$

Thus, the closed-loop position errors satisfy (11) if the parametric condition (14) holds. ■

Existence of parameter values that satisfy (14) can be established by examining convergent geometric series. For simplicity, consider setting $\beta_i = \beta \in (0,1)$, and $\varepsilon_{i,k} = \varepsilon_{1,k} = \varepsilon^k$, with $\varepsilon \in (0,1)$, for all $i = 1,\dots,N_v$. Then

$$\beta_i + \sum_{k=1}^{\infty} \varepsilon_{i,k}(1+\varepsilon_{1,k}) = \beta + \frac{1}{1-\varepsilon} + \frac{1}{1-\varepsilon^2} - 2$$

using the familiar series relation $\sum_{n=0}^{\infty} x^n = 1 + x + x^2 + \ldots = 1/(1-x)$, provided $|x| < 1$. Since the sums on ε start with an index of $k = 1$, we removed the x^0 term, and applied this relation to $x = \varepsilon$ and $x = \varepsilon^2$ separately in the expression above. Thus, we must choose (β,ε) to satisfy

$$\beta + \frac{1}{1-\varepsilon} + \frac{1}{1-\varepsilon^2} = \beta + \frac{2+\varepsilon}{1-\varepsilon^2} < 3$$

with the added conditions that $\beta, \varepsilon \in (0,1)$. Example values that satisfy this include $(\beta,\varepsilon) = (0.7, 0.2)$, (0.87, 0.1) or 0.45, 0.3). As stated, Assumption IV.1 is required in Method 1 for these parametric conditions to result in leader-follower string stability.

2) Method 2: For the method developed here, we show that for any $j = 2,\dots,N_v$ there exists an $\alpha_j \in (0,1)$ such that the closed-loop position error satisfies

$$\left|q_{j,\mathrm{e}}^*(t;t_k)\right|_{\infty,\delta} \le \alpha_j \max_{l\in\{0,\dots,k\}}\left|q_{1,\mathrm{e}}^*(t;t_l)\right|_{\infty,\delta}, \ \forall k = 0,1,2,\dots \tag{21}$$

Equation (21) is a sufficient condition that ensures (3), and removes the need for Assumption IV.1.

Algorithm IV.2: Distributed receding horizon controller for any vehicle $i \in \{1,\dots,N_v\}$:

Data: $T \in (0,\infty)$, $\delta \in (0,T]$. Platoon is at equilibrium ($z_i = 0$, $\forall i$) for all time $t < 0$.

Initialization with update period $[-\delta, 0]$: At some time $t' < 0$ $-\delta$, the lead car $i = 1$ broadcasts the intent to change desired speed to a new constant value, starting at time $t_0 = 0$. All cars receive the new desired speed $\dot{q}_L$. In addition, during the initialization update period $[-\delta, 0]$:

1) Car 1 solves Problem III.1 setting $F_1 = 0$ and using the new desired speed, then broadcasts $q_{1,\mathrm{e}}^*(\cdot;t_0)$ to all cars $2,\dots,N_v$ in the platoon. Next:
2) Each car $i \ge 2$ solves Problem III.1 with the addition constraint:

$$\left|q_{i,\mathrm{e}}^p(t;t_0)\right| \le \beta_i\left|q_{1,\mathrm{e}}^*(t;t_0)\right|, \quad \forall t \in [t_0, t_0+T] \tag{22}$$

setting $\beta_i \in (0,1)$, $F_i = G_i = 0$ and using the new desired speed.

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

1) Apply $u_i^*(\tau;t_k)$, $\tau \in [t_k, t_{k+1})$.
2) Compute predicted state $z_i(t_{k+1}) = z_i^*(t_{k+1};t_k)$.

3) Compute $\hat{z}_i(\tau; t_{k+1})$ as defined by (7).
4) Transmit $\hat{z}_i(\cdot; t_{k+1})$ to follower $i+1$ and for $j \geq 2$ receive $\hat{z}_j(\cdot; t_{k+1})$ from predecessor $j = i - 1$. Additionally, all $i \geq 2$ receive $\hat{q}_{1,\mathrm{e}}(\cdot; t_{k+1})$ from the lead car.
5) Solve Problem III.1 for $u_i^*(\cdot; t_{k+1})$, with an additional constraint. For car $i = 1$:

$$\left|q^p_{i,\mathrm{e}}(t; t_{k+1}) - \hat{q}_{i,\mathrm{e}}(t; t_{k+1})\right|_\infty \leq \varepsilon_{i,k+1} \left|q^p_{1,\mathrm{e}}(t; t_{k+1})\right|_{\infty,\delta} \tag{23}$$

and for car $i \geq 2$:

$$\left|q^p_{i,\mathrm{e}}(t; t_{k+1}) - \hat{q}_{i,\mathrm{e}}(t; t_{k+1})\right|_\infty \leq \varepsilon_{i,k+1} \left|\hat{q}_{1,\mathrm{e}}(t; t_{k+1})\right|_{\infty,\delta} \tag{24}$$

setting the update-dependent parameter $\varepsilon_{i,k+1} > 0$.

Observe that (22) is a harder constraint to satisfy than (12). Constraints (23) and (24) are generally harder to satisfy than (13) also.

Theorem 3: Suppose Assumption II.1, the conditions of Lemma IV.2 hold, and that in the application of Algorithm IV.2 a feasible solution $(z_i^*(\cdot; t_0), u_i^*(\cdot; t_0))$ is found at initialization for every $i = 1, \ldots, N_v$. Then, by application of Algorithm IV.2:

1) The assumed trajectories $(z_i^p(\cdot; t_k), u_i^p(\cdot; t_k)) = (\hat{z}_i(\cdot; t_k), \hat{u}_i(\cdot; t_k))$ are a feasible solution at every subsequent update $k \geq 1$, and for every $i = 1, \ldots, N_v$;
2) The stability properties of Algorithm III.1 via Theorem 1 are preserved.
3) The closed-loop position errors satisfy (21) if the following parametric condition holds, $\forall i = 2, \ldots, N_v$:

$$\beta_i + \beta_i \sum_{k=1}^{\infty} \varepsilon_{1,k} + \sum_{k=1}^{\infty} \varepsilon_{i,k}(1 + \varepsilon_{1,k}) < 1. \tag{25}$$

Proof: Parts 1 and 2 follow Theorem 2. From (22) at initialization, for the $k = 0$ case we have

$$\left|q^*_{i,\mathrm{e}}(t; t_0)\right|_{\infty,\delta} \leq \beta_i \left|q^*_{1,\mathrm{e}}(t; t_0)\right|_{\infty,\delta}. \tag{26}$$

Consider the case $k = 1$ next. From the triangle inequality and (24), for any $i = 2, \ldots, N_v$, we have on the interval $[t_1, t_1 + T]$

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t; t_1)\right| &\leq \left|q^*_{i,\mathrm{e}}(t; t_1) - \hat{q}_{i,\mathrm{e}}(t; t_1)\right| + \left|\hat{q}_{i,\mathrm{e}}(t; t_1)\right| \\ &\leq \varepsilon_{i,1} \left|\hat{q}_{1,\mathrm{e}}(t; t_1)\right|_{\infty,\delta} + \left|\hat{q}_{i,\mathrm{e}}(t; t_1)\right| \end{aligned} \tag{27}$$

From the triangle inequality and (23) for lead car $i = 1$,

$$\left|\hat{q}_{1,\mathrm{e}}(t; t_1)\right| \leq \varepsilon_{1,1} \left|q^*_{1,\mathrm{e}}(t; t_1)\right|_{\infty,\delta} + \left|q^*_{1,\mathrm{e}}(t; t_1)\right|. \tag{28}$$

for all $t \in [t_1, t_1 + T]$. This implies $|\hat{q}_{1,\mathrm{e}}(t; t_1)|_{\infty,\delta} \leq (1 + \varepsilon_{1,1})|q^*_{1,\mathrm{e}}(t; t_1)|_{\infty,\delta}$. From (22) at initialization, $|\hat{q}_{i,\mathrm{e}}(t; t_1)| \leq \beta_i |\hat{q}_{1,\mathrm{e}}(t; t_1)|$ for all $t \in [t_1, t_1 + T]$. Combining these two last bounds with (27) gives

$$\left|q^*_{i,\mathrm{e}}(t;t_1)\right| \leq (\varepsilon_{i,1}(1+\varepsilon_{1,1}) + \beta_i\varepsilon_{1,1})\left|q^*_{1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} + \beta_i\left|q^*_{1,\mathrm{e}}(t;t_1)\right| \tag{29}$$

for all $t \in [t_1, t_1 + T]$. This inequality results in a bound for the $k = 1$ case as

$$\left|q^*_{i,\mathrm{e}}(t; t_1)\right|_{\infty,\delta} \leq (\beta_i + \varepsilon_{i,1}(1+\varepsilon_{1,1}) + \beta_i\varepsilon_{1,1}) \left|q^*_{1,\mathrm{e}}(t; t_1)\right|_{\infty,\delta}. \tag{30}$$

The form of (29) over the entire interval $[t_1, t_1 + T]$ is required next, in addressing the $k = 2$ case. Following the initial steps for (27) and (28) above, for any $i = 2, \ldots, N_v$, we have on the interval $[t_2, t_2 + T]$

$$\left|q^*_{i,\mathrm{e}}(t; t_2)\right| \leq \varepsilon_{i,2}(1 + \varepsilon_{1,2}) \left|q^*_{1,\mathrm{e}}(t; t_2)\right|_{\infty,\delta} + \left|\hat{q}_{i,\mathrm{e}}(t; t_2)\right| \tag{31}$$

From (29), the triangle inequality, and (23),

$$\begin{aligned}\left|\hat{q}_{i,\mathrm{e}}(t; t_2)\right| &\leq (\varepsilon_{i,1}(1 + \varepsilon_{1,1}) + \beta_i\varepsilon_{1,1}) \left|q^*_{1,\mathrm{e}}(t; t_1)\right|_{\infty,\delta} \\ &\quad + \beta_i \left|\hat{q}_{1,\mathrm{e}}(t; t_2)\right| \\ &\leq (\varepsilon_{i,1}(1 + \varepsilon_{1,1}) + \beta_i\varepsilon_{1,1}) \left|q^*_{1,\mathrm{e}}(t; t_1)\right|_{\infty,\delta} \\ &\quad + \beta_i\varepsilon_{1,2} \left|q^*_{1,\mathrm{e}}(t; t_2)\right|_{\infty,\delta} + \beta_i \left|q^*_{1,\mathrm{e}}(t; t_2)\right| \end{aligned}$$

for all $t \in [t_2, t_2 + T]$. Combining this bound with (31) gives

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t; t_2)\right| &\leq (\beta_i(\varepsilon_{1,1} + \varepsilon_{1,2}) + \varepsilon_{i,1}(1 + \varepsilon_{1,1}) + \varepsilon_{i,2}(1 + \varepsilon_{1,2})) \\ &\quad \times \max_{l=1,2} \left|q^*_{1,\mathrm{e}}(t; t_l)\right|_{\infty,\delta} + \beta_i \left|q^*_{1,\mathrm{e}}(t; t_2)\right| \end{aligned} \tag{32}$$

for all $t \in [t_2, t_2 + T]$. This inequality results in a bound for the $k = 2$ case as

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t; t_2)\right|_{\infty,\delta} &\leq (\beta_i + \beta_i(\varepsilon_{1,1} + \varepsilon_{1,2}) + \varepsilon_{i,1}(1 + \varepsilon_{1,1}) \\ &\quad + \varepsilon_{i,2}(1 + \varepsilon_{1,2})) \max_{l=1,2} \left|q^*_{1,\mathrm{e}}(t; t_l)\right|_{\infty,\delta}. \end{aligned} \tag{33}$$

Carrying this out recursively, we have the $k = n$ case:

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t; t_n)\right|_{\infty,\delta} &\leq \left(\beta_i + \beta_i \sum_{j=1}^{n} \varepsilon_{1,j} + \sum_{j=1}^{n} \varepsilon_{i,j}(1 + \varepsilon_{1,j})\right) \\ &\quad \times \max_{l=0,\ldots,n} \left|q^*_{1,\mathrm{e}}(t; t_l)\right|_{\infty,\delta}. \end{aligned}$$

Thus, the closed-loop position errors satisfy (21) if the parametric condition (25) holds. ■

As before, existence of parameter values that satisfy (25) can be established by examining convergent geometric series. Setting $\beta_i = \beta \in (0, 1)$, and $\varepsilon_{i,k} = \varepsilon_{1,k} = \varepsilon^k$, with $\varepsilon \in (0, 1)$, for all $i = 1, \ldots, N_v$, we have

$$\begin{aligned}&\beta_i + \beta_i \sum_{k=1}^{\infty} \varepsilon_{1,k} + \sum_{k=1}^{\infty} \varepsilon_{i,k}(1 + \varepsilon_{1,k}) \\ &\quad = \beta + \beta \left[\frac{1}{1 - \varepsilon} - 1\right] + \frac{1}{1 - \varepsilon} + \frac{1}{1 - \varepsilon^2} - 2 \\ &\quad = \frac{\beta + 1}{1 - \varepsilon} + \frac{1}{1 - \varepsilon^2} - 2 \end{aligned}$$

We must therefore choose (β, ε) to satisfy

$$\frac{\beta + 1}{1 - \varepsilon} + \frac{1}{1 - \varepsilon^2} < 3$$

with the added conditions that $\beta, \varepsilon \in (0, 1)$. An example set of values that satisfy this include $(\beta, \varepsilon) = (0.55, 0.2)$, which is a reduced value in β compared to the values (0.7, 0.2) that satisfied (14). Although the parametric conditions on $\{\beta_i, \varepsilon_{i,j}\}$ are tighter, and the constraints in the modified optimal control problems are (moderately) more restrictive, Assumption IV.1 is no longer required for leader-follower string stability.

V. Predecessor-Follower String Stability

In the analysis here, we redefine the local optimal control problems and algorithm in a way that is comparable to Method 2 above. Our goal is to show that for any $j = 2, \dots, N_v$ there exists an $\alpha_j \in (0,1)$ such that the closed-loop position error satisfies

$$\left|q^*_{j,\mathrm{e}}(t;t_k)\right|_{\infty,\delta} \le \alpha_j \max_{l\in\{0,\dots,k\}} \left|q^*_{j-1,\mathrm{e}}(t;t_l)\right|_{\infty,\delta}, \quad \forall k \ge 0. \tag{34}$$

Equation (34) is a sufficient condition that ensures (4). In the algorithm below, cars $i \ge 3$ need to communicate with the lead car $i = 1$ *only at initialization*. This is an advantage over string-stable controllers that require continuous position, speed or acceleration information from the lead and preceding cars.

Algorithm IV.3: Distributed receding horizon controller for any vehicle $i \in \{1, \dots, N_v\}$:

Data: $T \in (0, \infty)$, $\delta \in (0, T]$. Platoon is at equilibrium ($z_i = 0$, $\forall i$) for all time $t < 0$.

Initialization with update period $[-\delta, 0]$: At some time $t' < 0$ $-\delta$, the lead car $i = 1$ broadcasts the intent to change desired speed to a new constant value, starting at time $t_0 = 0$. All cars receive the new desired speed $\dot{q}_L$. In addition, during the initialization update period $[-\delta, 0]$.

1) Car 1 solves Problem III.1 setting $F_1 = 0$ and using the new desired speed, then broadcasts $q^*_{1,\mathrm{e}}(\cdot;t_0)$ to all cars $2, \dots, N_v$ in the platoon. Next:
2) Each car $i \ge 2$ solves Problem III.1 with the addition constraint:

$$(1-\xi_i)\gamma_i \left|q^*_{1,\mathrm{e}}(t;t_0)\right| \le \left|q^p_{i,\mathrm{e}}(t;t_0)\right| \le (1+\xi_i)\gamma_i \left|q^*_{1,\mathrm{e}}(t;t_0)\right| \tag{35}$$

for all $t \in [t_0, t_0 + T]$, setting $\gamma_i, \xi_i \in (0,1)$, $F_i = G_i = 0$ and using the new desired speed.

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

1) Apply $u^*_i(\tau; t_k)$, $\tau \in [t_k, t_{k+1})$.
2) Compute predicted state $z_i(t_{k+1}) = z^*_i(t_{k+1}; t_k)$.
3) Compute $\hat{z}_i(\tau; t_{k+1})$ as defined by (7).
4) Transmit $\hat{z}_i(\cdot; t_{k+1})$ to follower $i+1$ and for $j \ge 2$ receive $\hat{z}_j(\cdot; t_{k+1})$ from predecessor $j = i - 1$.
5) Solve Problem III.1 for $u^*_i(\cdot; t_{k+1})$, with an additional constraint. For car $i = 1$ the added constraint is (23) as before. For car $i = 2, \dots, N_v - 1$:

$$\begin{aligned}&\left|q^p_{i,\mathrm{e}}(t;t_{k+1}) - \hat{q}_{i,\mathrm{e}}(t;t_{k+1})\right|_\infty \\ &\quad \le \varepsilon_{i,k+1} \min\left\{\left|\hat{q}_{i-1,\mathrm{e}}(t;t_{k+1})\right|_{\infty,\delta}, \left|q^p_{i,\mathrm{e}}(t;t_{k+1})\right|_{\infty,\delta}\right\}\end{aligned} \tag{36}$$

setting the update-dependent and location-within-platoon-dependent parameter $\varepsilon_{i,k+1} > 0$. For car $i = N_v$, the constraint is (36) but setting the right-hand side to $\varepsilon_{N_v,k+1}|\hat{q}_{N_v-1,\mathrm{e}}(t;t_{k+1})|_{\infty,\delta}$.

Before establishing string stability, we analyze the initialization portion of our algorithm. By design, the form of (35) is a distributed way to construct a trajectory that is initially predecessor-follower string stable. We show this in the following lemmas. The first lemma shows how the predecessor-follower error gain β_i for each $i = 2, \dots, N_v$ is defined by a subset of the design parameters $\{\xi_i, \gamma_i\}_{i=2,\dots,N_v}$.

Lemma IV.3: By the distributed initialization defined by (35) in Algorithm IV.3, for all $i = 2, \dots, N_v$, the initial trajectories satisfy

$$\left|q^*_{i,\mathrm{e}}(t;t_0)\right| \le \beta_i \left|q^*_{i-1,\mathrm{e}}(t;t_0)\right|, \quad \forall t \in [t_0, t_0 + T] \tag{37}$$

with $\beta_2 = (1+\xi_2)\gamma_2$, $\beta_i = ((1+\xi_i)/(1-\xi_{i-1}))(\gamma_i/\gamma_{i-1})$, $i = 3, \dots, N_v$.

Proof: From (35), we have lower and upper bounds for i and $i-1$ as

$$(1-\xi_i)\gamma_i \left|q^*_{1,\mathrm{e}}(t;t_0)\right| \le \left|q^*_{i,\mathrm{e}}(t;t_0)\right| \le (1+\xi_i)\gamma_i \left|q^*_{1,\mathrm{e}}(t;t_0)\right|$$

and

$$\begin{aligned}(1-\xi_{i-1})\gamma_{i-1} \left|q^*_{1,\mathrm{e}}(t;t_0)\right| &\le \left|q^*_{i-1,\mathrm{e}}(t;t_0)\right| \\ &\le (1+\xi_{i-1})\gamma_{i-1} \left|q^*_{1,\mathrm{e}}(t;t_0)\right|\end{aligned}$$

for all $t \in [t_0, t_0 + T]$. Using the upper bound on $|q^*_{i,\mathrm{e}}(t;t_0)|$ and lower-bound on $|q^*_{i-1,\mathrm{e}}(t;t_0)|$, (37) follows. ■

Next, one must choose a parameter set $\{\xi_i, \gamma_i\}_{i=2,\dots,N_v}$ such that β_i defined in (37) satisfies $\beta_i \in (0,1)$. In one approach, each car i could choose a value for $\beta_i \in (0,1)$, then given values from its predecessor ξ_{i-1}, γ_{i-1}, chooses ξ_i, γ_i using $(1+\xi_i)\gamma_i = \beta_i(1-\xi_{i-1})\gamma_{i-1}$. For car 2, this simplifies to $(1+\xi_2)\gamma_2 = \beta_2$. Alternatively, the following lemma defines a parameter set for the case $\beta_i = \rho \in (0,1)$ for all $i = 2, \dots, N_v$, which may be desirable in some platooning situations.

Lemma IV.4: Let $\epsilon, \rho \in (0,1)$. Then, a parameter set $\{\xi_i, \gamma_i\}_{i=2,\dots,N_v}$ that satisfies $\beta_i = \rho \in (0,1)$, for all $i = 2, \dots, N_v$ is given by

$$\xi_i = \epsilon^{i-1}, \quad \gamma_i = \frac{\rho^{i-1}}{1+\epsilon^{i-1}} \prod_{j=1}^{i-2} \frac{1-\epsilon^j}{1+\epsilon^j}, \qquad i = 2, \dots, N_v. \tag{38}$$

Proof: Substitution from (38) into the definition for each β_i shows that $\beta_i = \rho$. In particular, observe that $\gamma_2 = \rho/(1+\epsilon)$ and $\gamma_{i+1} = \gamma_i \rho(1-\epsilon^{i-1})/(1+\epsilon^i)$, for $i \ge 2$. ■

Existence of $\{\xi_i, \gamma_i\}_{i=2,\dots,N_v}$ values for large N_v is an issue to consider. For the constructed set of parameters in Lemma IV.4, ξ_n and γ_n get smaller as n increases, which means the position errors down the string must be closer to zero than their predecessors. The initialization constraint (35) used to generate initially string-stable trajectories can be re-written equivalently as

$$\left\|q^p_{i,\mathrm{e}}(t;t_0)\right| - \gamma_i\left|q^*_{1,\mathrm{e}}(t;t_0)\right\| \le \xi_i\gamma_i \left|q^*_{1,\mathrm{e}}(t;t_0)\right|, \ \forall t \in [t_0, t_0 + T]. \tag{39}$$

The combined parameter $\xi_n\gamma_n$ acts as a gain in (39) that affects the allowable deviation of $q^p_{i,\mathrm{e}}(t;t_0)$ from $\gamma_i q^*_{1,\mathrm{e}}(t;t_0)$ for all $t \in [t_0, t_0 + T]$. Using (38), values for ξ_n, γ_n and combined parameter $\xi_n\gamma_n$ versus the vehicle number n is shown in Fig. 1 for up to a platoon size of $N_v = 10$ vehicles, setting $\rho = 0.99$

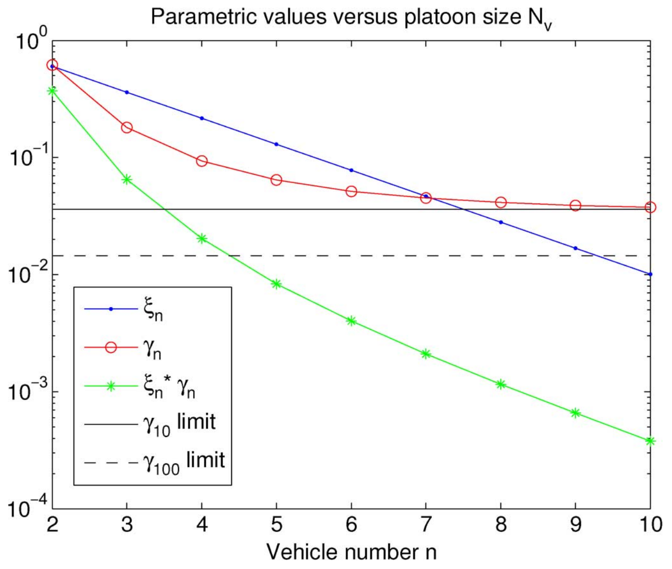


Fig. 1. Values for ξ_n, γ_n and combined parameter $\xi_n \gamma_n$ versus vehicle number n, using (38). The plots is shown for up to ten vehicles and setting $\rho = 0.99$ and $\epsilon = 0.6$, parameters defined in Lemma IV.4. The γ_{10} and γ_{100} limits were computed as described in the text.

and $\epsilon = 0.6$. Practically, the parameter values and implemented constraints only make sense for finite N_v. For $N_v \to \infty$, $\xi_n \to 0$ exponentially in $n = N_v$. The trend for γ_n can be examined, first by rewriting it as

$$\gamma_{n+1} = \frac{\rho^n}{1-\epsilon^n} \prod_{l=1}^{n-1} \frac{1-\left(\frac{1}{\epsilon}\right)^{-l}}{1+\left(\frac{1}{\epsilon}\right)^{-l}}.$$

In the limit of large N_v,

$$\prod_{k=1}^{\infty} \frac{1-\left(\frac{1}{\epsilon}\right)^{-k}}{1+\left(\frac{1}{\epsilon}\right)^{-k}} = 1 + 2\sum_{k=1}^{\infty}(-1)^k \epsilon^{k^2} =: \theta_4(0,\epsilon) \tag{40}$$

where θ_4 is the Jacobi theta function [24]. Up to any finite N_v, we can observe the convergent behavior of γ_n using the first several terms in the summation in (40) (summing up to N_v, for example) and multiplying it by $\rho^{N_v}/(1-\epsilon^{N_v})$. We do this to compute γ_{10} and γ_{100} shown in Fig. 1. Observe from the figure that $\xi_n \gamma_n$, the gain parameter in the constraint (39), decreases dramatically, forcing $q^p_{n,\mathrm{e}}(t;t_0) \to \gamma_n q^*_{1,\mathrm{e}}(t;t_0)$. In this analysis, the parametric conditions for initialization were derived to ensure $\beta_i = \rho \in (0,1)$, for all $i = 2,\ldots,N_v$; however, an alternative set that permits β_i to be smaller as i increases, for example, could alleviate the rate of decrease of $\xi_n \gamma_n$ with n.

Accepting the limitations of our initialization method, we now present the final theoretical result of the paper, and demonstrate that Algorithm IV.3 ensure predecessor-follower string stability. For this purpose, constraint (36) essentially combines (23) and (24), as each vehicle $i = 2,\ldots,N_v - 1$ acts as a predecessor (local leader) and a follower. Since $i = N_v$ is strictly a follower, (36) does not require the $\min\{\cdot\}$ on the right-hand side.

Theorem 4: Suppose Assumption II.1, the conditions of Lemma IV.2 hold, and that in the application of Algorithm IV.3 a feasible solution $(z^*_i(\cdot;t_0), u^*_i(\cdot;t_0))$ is found at initialization for every $i = 1,\ldots,N_v$. Then, by application of Algorithm IV.3:

1) The assumed trajectories $(z^p_i(\cdot;t_k), u^p_i(\cdot;t_k)) = (\hat{z}_i(\cdot;t_k), \hat{u}_i(\cdot;t_k))$ are a feasible solution at every subsequent update $k \geq 1$, and for every $i = 1,\ldots,N_v$;
2) The stability properties of Algorithm III.1 via Theorem 1 are preserved.
3) The closed-loop position errors satisfy (34) if the following parametric condition holds, $\forall i = 2,\ldots,N_v$:

$$\beta_i + \beta_i \sum_{k=1}^{\infty} \varepsilon_{i-1,k} + \sum_{k=1}^{\infty} \varepsilon_{i,k}(1+\varepsilon_{i-1,k}) < 1. \tag{41}$$

Proof: Parts 1 and 2 follow from the proof of Theorem 2. It remains to prove part 3. From initialization and Lemma IV.3, for $i = 2,\ldots,N_v$ and the $k = 0$ case we have

$$\left|q^*_{i,\mathrm{e}}(t;t_0)\right|_{\infty,\delta} \leq \beta_i \left|q^*_{i-1,\mathrm{e}}(t;t_0)\right|_{\infty,\delta}. \tag{42}$$

Consider the case $k = 1$ next. From the triangle inequality, and (36) for any $i = 2,\ldots,N_v$, we have on the interval $[t_1, t_1 + T]$

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t;t_1)\right| &\leq \left|q^*_{i,\mathrm{e}}(t;t_1) - \hat{q}_{i,\mathrm{e}}(t;t_1)\right| + \left|\hat{q}_{i,\mathrm{e}}(t;t_1)\right| \\ &\leq \varepsilon_{i,1}\left|\hat{q}_{i-1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} + \left|\hat{q}_{i,\mathrm{e}}(t;t_1)\right| \end{aligned} \tag{43}$$

From the triangle inequality and (23) for lead car and (36) for all predecessor cars, for any $i - 1 = 1,\ldots,N_v - 1$,

$$\left|\hat{q}_{i-1,\mathrm{e}}(t;t_1)\right| \leq \varepsilon_{i-1,1}\left|q^*_{i-1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} + \left|q^*_{i-1,\mathrm{e}}(t;t_1)\right| \tag{44}$$

for all $t \in [t_1, t_1 + T]$. This implies $|\hat{q}_{i-1,\mathrm{e}}(t;t_1)|_{\infty,\delta} \leq (1+\varepsilon_{i-1,1})|q^*_{i-1,\mathrm{e}}(t;t_1)|_{\infty,\delta}$. From Lemma IV.3, $|\hat{q}_{i,\mathrm{e}}(t;t_1)| \leq \beta_i|\hat{q}_{i-1,\mathrm{e}}(t;t_1)|$ for all $t \in [t_1, t_1 + T]$. Combining the last two bounds with (43) gives

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t;t_1)\right| \leq (\varepsilon_{i,1}(1+\varepsilon_{i-1,1}) + \beta_i\varepsilon_{i-1,1})\left|q^*_{i-1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} \\ +\beta_i\left|q^*_{i-1,\mathrm{e}}(t;t_1)\right| \end{aligned} \tag{45}$$

for all $t \in [t_1, t_1 + T]$. This inequality results in a bound for the $k = 1$ case as

$$\begin{aligned}\left|q^*_{i,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} \leq (\beta_i + \varepsilon_{i,1}(1+\varepsilon_{i-1,1}) + \beta_i\varepsilon_{i-1,1}) \\ \times\left|q^*_{i-1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta}. \end{aligned} \tag{46}$$

Following the steps for (43) and (44) above, for any $i = 2,\ldots,N_v$, we have on the interval $[t_2, t_2 + T]$

$$\left|q^*_{i,\mathrm{e}}(t;t_2)\right| \leq \varepsilon_{i,2}(1+\varepsilon_{i-1,2})\left|q^*_{i-1,\mathrm{e}}(t;t_2)\right|_{\infty,\delta} + \left|\hat{q}_{i,\mathrm{e}}(t;t_2)\right|. \tag{47}$$

From (45), (23), and (36), we have

$$\begin{aligned}\left|\hat{q}_{i,\mathrm{e}}(t;t_2)\right| &\leq (\varepsilon_{i,1}(1+\varepsilon_{i-1,1}) + \beta_i\varepsilon_{i-1,1})\left|q^*_{i-1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} \\ &\quad + \beta_i\left|\hat{q}_{i-1,\mathrm{e}}(t;t_2)\right| \\ &\leq (\varepsilon_{i,1}(1+\varepsilon_{i-1,1}) + \beta_i\varepsilon_{i-1,1})\left|q^*_{i-1,\mathrm{e}}(t;t_1)\right|_{\infty,\delta} \\ &\quad + \beta_i\varepsilon_{i-1,2}\left|q^*_{i-1,\mathrm{e}}(t;t_2)\right|_{\infty,\delta} + \beta_i\left|q^*_{i-1,\mathrm{e}}(t;t_2)\right| \end{aligned}$$

for all $t \in [t_2, t_2 + T]$. Combining this bound with (47) gives

$$\left|q^*_{i,\mathrm{e}}(t;t_2)\right| \le \beta_i \left|q^*_{i-1,\mathrm{e}}(t;t_2)\right| + \max_{l=1,2} \left|q^*_{i-1,\mathrm{e}}(t;t_l)\right|_{\infty,\delta} \times \left(\beta_i(\varepsilon_{i-1,1} + \varepsilon_{i-1,2}) + \sum_{j=1}^{2} \varepsilon_{i,j}(1+\varepsilon_{i-1,j}) \right)$$

for all $t \in [t_2, t_2 + T]$. This inequality results in a bound for the $k = 2$ case as

$$\left|q^*_{i,\mathrm{e}}(t;t_2)\right|_{\infty,\delta} \le \max_{l=1,2} \left|q^*_{i-1,\mathrm{e}}(t;t_l)\right|_{\infty,\delta} \times \left(\beta_i + \beta_i(\varepsilon_{i-1,1} + \varepsilon_{i-1,2}) + \sum_{j=1}^{2} \varepsilon_{i,j}(1+\varepsilon_{i-1,j}) \right). \quad (48)$$

Carrying this out recursively, we have the $k = n$ case

$$\left|q^*_{i,\mathrm{e}}(t;t_n)\right|_{\infty,\delta} \le \max_{l=0,\dots,n} \left|q^*_{i-1,\mathrm{e}}(t;t_l)\right|_{\infty,\delta} \times \left(\beta_i + \beta_i \sum_{k=1}^{n} \varepsilon_{i-1,k} + \sum_{k=1}^{n} \varepsilon_{i,k}(1+\varepsilon_{i-1,k}) \right).$$

Thus, the closed-loop position errors satisfy (34) if the parametric condition (41) holds. ■

Existence of parameter values that satisfy (41) when all vehicles share identical $\beta_i = \beta \in (0,1)$, and $\varepsilon_{i,k} = \varepsilon^k$, with $\varepsilon \in (0,1)$, was addressed following the proof of Theorem 3. Recall that the choice of parameters $\{\xi_i, \gamma_i\}_{i=2,\dots,N_v}$ affects the value for β_i from the initialization phase, as defined in (37). The result in Lemma IV.4 applies to the case when all vehicles share identical $\beta_i = \beta \in (0,1)$, and so connects with the analysis following the proof of Theorem 3.

For Predecessor-Follower string stability, we now analyze the case when each vehicle chooses its own values for $\{\beta_i, \varepsilon_i\}$. As stated following Lemma IV.4, an independent choice by vehicle i for β_i requires values $\{\xi_{i-1}, \gamma_{i-1}\}$ from its predecessor, and selection of $\{\xi_i, \gamma_i\}$ using $(1+\xi_i)\gamma_i = \beta_i(1-\xi_{i-1})\gamma_{i-1}$. In what follows, we further simplify the choice of parameters by setting $\xi_i = \varepsilon_i$, and so $\{\beta_i, \varepsilon_i\}$ are the only variables chosen by vehicle i. These two variables specify the tradeoff made by vehicle i between the attenuation of its preceding vehicle's positioning error and its own move suppression. Now, setting $\varepsilon_{i,k} = \varepsilon_i^k$ in (41) gives

$$\begin{aligned} &\beta_i + \beta_i \sum_{k=1}^{\infty} \varepsilon_{i-1,k} + \sum_{k=1}^{\infty} \varepsilon_{i,k}(1+\varepsilon_{i-1,k}) \\ &\quad = \beta_i + \beta_i \left[\frac{1}{1-\varepsilon_{i-1}} - 1 \right] + \frac{1}{1-\varepsilon_i} + \frac{1}{1-\varepsilon_{i-1}\varepsilon_i} - 2 \\ &\quad = \frac{\beta_i}{1-\varepsilon_{i-1}} + \frac{1}{1-\varepsilon_i} + \frac{1}{1-\varepsilon_{i-1}\varepsilon_i} - 2. \end{aligned}$$

Therefore, given ε_{i-1} from its predecessor, each vehicle chooses $\{\beta_i, \varepsilon_i\} \in (0,1)^2$ to satisfy

$$\frac{\beta_i}{1-\varepsilon_{i-1}} + \frac{1}{1-\varepsilon_i} + \frac{1}{1-\varepsilon_{i-1}\varepsilon_i} < 3. \quad (49)$$

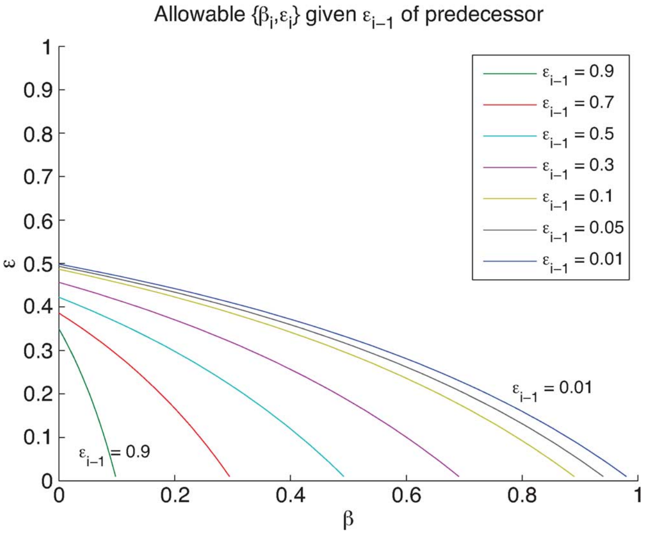


Fig. 2. Curves showing the limiting feasible values for $\{\beta_i, \varepsilon_i\}$ that satisfy (49) at equality for various values of ε_{i-1}. Values below each curve also satisfy (49).

Fig. 2 shows values of $\{\beta_i, \varepsilon_i\}$ that satisfy (49) at equality for various values of ε_{i-1}. Thus, all values below each curve satisfy the inequality (49).

Each vehicle computes its values of $\{\beta_i, \varepsilon_i\}$ upon receiving $\{\xi_{i-1} = \varepsilon_{i-1}, \gamma_{i-1}\}$ from its predecessor. Therefore, the choice of values by each vehicle must occur sequentially down the platoon at the time of platoon formation, as described in Section VI. The choices available to the vehicle will depend on the choices made by vehicles ahead of it in the platoon.

As stated, our condition of Predecessor-Follower string stability ensures Leader-Follower string stability. Although the constraint (36) has the computed trajectory on both sides of the inequality, the right-hand side could be simplified to be

$$\min\left\{ \left|\hat{q}_{i-1,\mathrm{e}}(t;t_{k+1})\right|_{\infty,\delta}, q_{i,\mathrm{e}}(t_{k+1}) \right\}$$

since $|q^p_{i,\mathrm{e}}(t;t_{k+1})|_{\infty,\delta} \ge q^p_{i,\mathrm{e}}(t_{k+1};t_{k+1})$, and string stability would still apply. The constraint (36) would then be linear in the computed trajectory $q^p_{i,\mathrm{e}}$, which would simplify its implementation.

VI. Simulations

The simulations explore tradeoffs in the move suppression F_i and predecessor-relative error G_i cost weights, and their affect on stability and string stability. The move suppression cost is a soft-form of the position error constraints used in the previous section to prove string stability. For a seven-car platoon simulation, the platoon starts from equilibrium and then at time $t = 1$ second, the lead car requires a desired speed increase from 25 to 26 m/sec. Thus, $t = 1$ is the "time $t = 0$" referred to in the definitions and algorithms presented previously. A horizon of $T = 5$ s and update period of $\delta = 0.5$ s are used. We consider the nonlinear (quadratic drag) error dynamics

$$m_i \ddot{q}_{i,\mathrm{e}} = u_i - c_i \dot{q}^2_{i,\mathrm{e}}$$

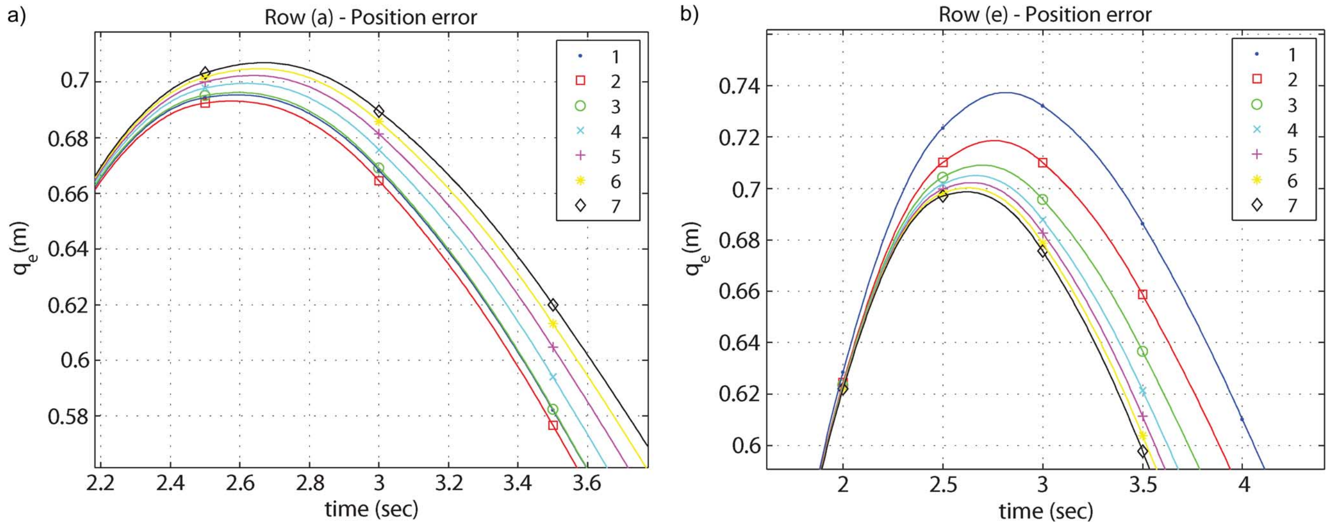


Fig. 3. Close-up of position errors trajectories $q_{i,\mathrm{e}}$ during their maximum values for parameter values corresponding to (a) a **string unstable** response [Table I, row (a)], and (b) **a leader-follower and predecessor-follower string stable** response [Table I, row (e)].

setting $m_i = 1841$ kg (4000 lbs) and $c_i = 0.41\ \mathrm{N/m}$. In the cost defined in Problem III.1, we have $Q_i = \mathrm{diag}(0.5, 1)$. Differential flatness in the error dynamics is used to formulate a discretized optimization problem for the variables $\{q_{i,\mathrm{e}}^p(t_j), \dot{q}_{i,\mathrm{e}}^p(t_j), \ddot{q}_{i,\mathrm{e}}^p(t_j)\}_{j=1,\ldots,M}$, with $u_i^p = m_i\ddot{q}_{i,\mathrm{e}} + c_i\dot{q}_{i,\mathrm{e}}^2$. Details on how flatness and the discretization are done are provided here [20]. The weight on the control is $R_i = 10^{-5}$, chosen so that the size of the control penalty (proportional to m_i^2) is comparable to the Q_i cost terms during the transient response. We employ the `fmincon` nonlinear programming problem solver in Matlab. The `fmincon` solver uses sequential quadratic programming; although solutions may be locally optimal, we accept this limitation for the purpose of illustrating our control algorithm. The table below summarizes results for different values for the matrices F_i and G_{i+1}, considering whether the sufficient condition for stability (9) in Lemma IV.2 is satisfied, and also if the definitions of leader-follower (L-F) and predecessor-follower (P-F) string stability are satisfied. Although rows (c)–(e) do not satisfy the sufficient conditions for stability in (9), all closed-loop responses converged and met the control objective. Interestingly, the L-F and P-F string stability margins improve from rows (b)–(d), that is, as the ratio G_{i+1}/F_i is increased, and row (e) exhibits the most string stable response. Rows (b)–(d) exhibit P-F gain convergence to $\alpha = 1$ for cars $i \geq 4$. On the other hand, scheduling the diagonal gain for G_{i+1} as $i+1$, as shown in row (e), maintains a P-F gain of less than 1 for all seven cars. Specifically, defining the gain $\alpha_{i,j} = \max_{t\geq 0}|q_{i,\mathrm{e}}(t)|/\max_{t\geq 0}|q_{j,\mathrm{e}}(t)|$ for $1 \leq i < j \leq N_v$, the L-F gains $(\alpha_{1,j})$ for row (e) are

$$(\alpha_{1,2}, \alpha_{1,3}, \alpha_{1,4}, \alpha_{1,5}, \alpha_{1,6}, \alpha_{1,7}) = (0.9746, 0.9617, 0.9561, 0.9524, 0.9497, 0.9475)$$

and the P-F gains $(\alpha_{i,i+1})$ are

$$(\alpha_{2,3}, \alpha_{3,4}, \alpha_{4,5}, \alpha_{5,6}, \alpha_{6,7}) = (0.98672,\ 0.99426,\ 0.99610,\ 0.99711,\ 0.99777).$$

Closed-loop position errors for rows (a) and (e) are zoomed near their maximum and plotted in Fig. 3. Closed-loop speed trajectories for rows (a) and (e) are zoomed and plotted in Fig. 4. Observe that the speed of the following cars "leads" that of the preceding cars, when string stability is satisfied [Fig. 4(b)]. This trend is consistent with the performance of other string stable controllers. The closed-loop acceleration trajectories $\ddot{q}_i$ are plotted in Fig. 5. Consistent with the speed trend in Fig. 4(b), the acceleration of vehicle 7 "exceeds" that of the preceding cars in the string stable response. Closer inspection of the response shows that the diamond (symbol for car 7 plotted at the update times) is further from the origin (more aggressive) during peak acceleration and deceleration. The acceleration for car 7 is also the first to converge to $\approx 0\ \mathrm{m/sec}^2$, showing the superior performance of the last car in the platoon. Simulations for row (e) in Table I took 9.36 s to execute for 20 receding horizon updates, with vehicle's update computed serially. This corresponds to an average of 67 ms per vehicle to execute steps 2–5 in the controller logic of Algorithm III.1, supporting the feasibility of real-time implementation using an update period of 0.5 s.

In other simulations, $G_i \to 0$ and $F_i \to F$, a constant, results in all vehicles exhibiting a closed-loop response that is consistent with the response computed at initialization. Thus, simulations suggest that a string stabilizing initial response could be preserved as such using the F_i costs alone. These trends are consistent with the form of the constraints (move suppression on the position error) used to prove L-F and P-F string stability in the theoretical sections.

VII. Platoon Formation

Constraints in the optimal control problems of the controllers (Algorithms II.1–IV.3) provided sufficient conditions for leader-follower and predecessor-follower string stability. Simulations in the previous section showed that these constraints are not necessary to achieve string stability. In this section and the next, we presume that the predecessor-follower

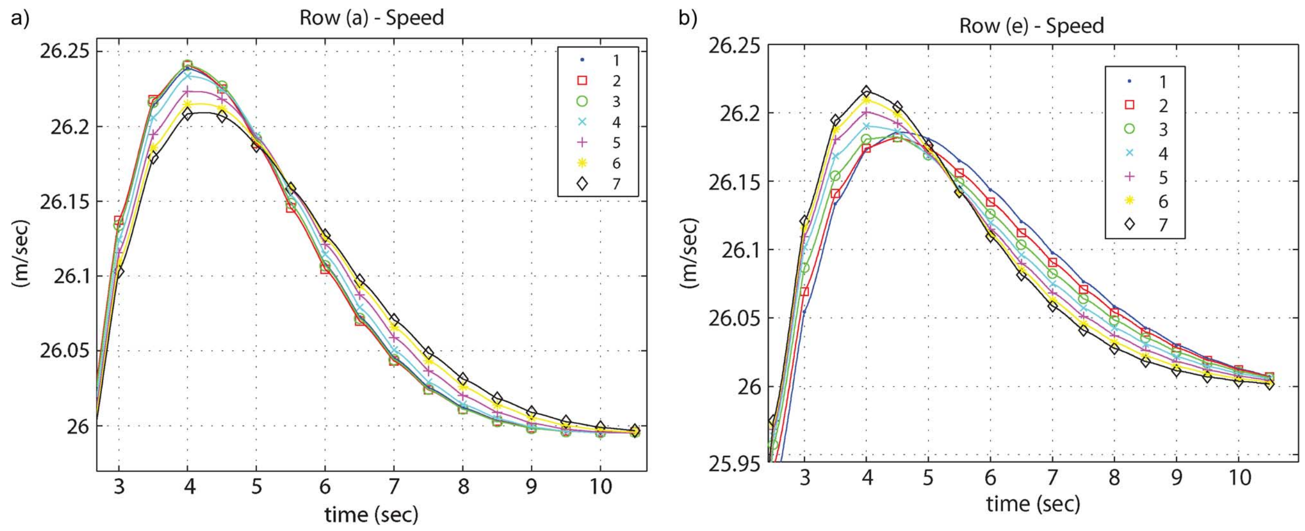


Fig. 4. Convergence of the closed-loop speed trajectories $\ddot{q}_i$ during the speed increase from 25 to 26 m/sec, again for parameter values corresponding to (a) Table I row (a), which is not string stable, and (b) Table I row (e) which is leader-follower and predecessor-follower string stable.

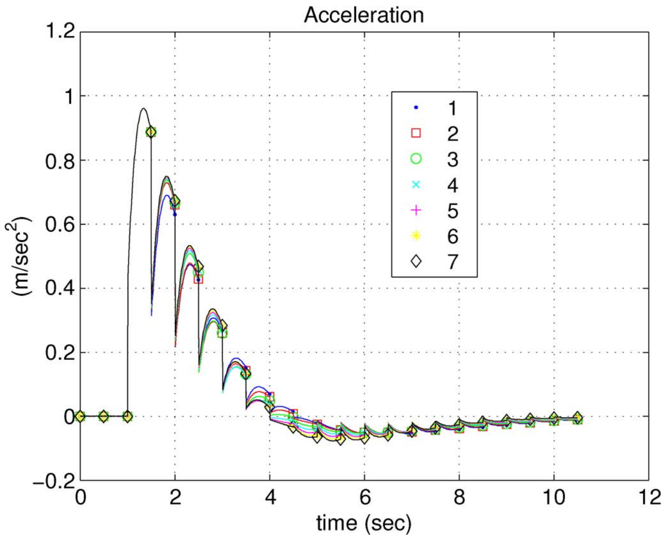


Fig. 5. Closed-loop acceleration trajectories $\ddot{q}_i$ before, during and after the speed increase from 25 to 26 m/sec at time $t = 1$ second. Parameter values correspond to the string stable case of row (e) in Table I.

TABLE I
STABILITY AND STRING STABILITY TRADE-OFFS FOR VARYING VALUES FOR MOVE SUPPRESSION AND PREDECESSOR-RELATIVE ERROR MATRICES

Row	$F_i{}^a$	G_{i+1}^a	satisfies (9)	L-F s.s.^{b}	P-F s.s.^{c}
(a)	$10/i$	$10/i$	Yes	No	No
(b)	10	10	Yes	Yes	Yes
(c)	1	20	No	Yes	Yes
(d)	0	50	No	Yes	Yes
(e)	0	$i+1$	No	Yes	Yes

[a] F_i and G_{i+1}, $i = 1, \ldots, N_v - 1$, are 2×2 diagonal matrices, with gains on the diagonal equal to the values shown.
[b] Leader-follower string stability.
[c] Predecessor-follower string stability.

string stable controller (Algorithm IV.3) is implemented, and examine practical aspects of platoon formation and size.

Before the first execution of Algorithm IV.3, a platoon that satisfies Theorem 4 must be formed by available cars, vans, and trucks. The choice of a leader is not discussed here and the order of the available cars behind this leader is assumed given. We remark only that, in light of our analysis below, the choice of a leader is not trivial because the leader will reap less reward from the platoon than other members; incentives for being a leader must be considered. To simplify the formation of the platoon, we again set $\xi_i = \varepsilon_i$ to allow computation of γ_i using (37) after $\{\beta_i, \varepsilon_i\}$ are chosen by vehicle i.

1) The leader chooses its epsilon ε_1 and communicates this value to the second vehicle. As shown in Section VII, a leader that chooses a small epsilon can lead longer platoons than a leader with a larger epsilon.
2) The second vehicle looks at Fig. 2 and chooses $\{\beta_2, \varepsilon_2\}$ that are good for its vehicle type (e.g., car, truck) and such that (41) holds for Predecessor-Follower stability (i.e., they are under the curve for the chosen $\varepsilon_{i-1} = \varepsilon_1$ in Fig. 2). If the second vehicle desires relaxed move suppression constraints, it will choose ε_2 as large as possible at the expense of β_2 being smaller. This will reduce γ_2 more than if the second vehicle chooses ε_2 as small as possible. However, relaxed move suppression constraints provide the vehicle increased flexibility to modify its trajectory based on its drive-train characteristics.
3) To ensure Predecessor-Follower stability, the second vehicle calculates $\gamma_2 = \beta_2/(1+\varepsilon_2)$ and passes $\{\varepsilon_2, \gamma_2\}$ to the third vehicle.
4) The third vehicle repeats step (2) using $\varepsilon_{i-1} = \varepsilon_2$.
5) To ensure Predecessor-Follower stability, the third vehicle calculates $\gamma_3 = \beta_3\gamma_2(1-\varepsilon_2)/(1+\varepsilon_3)$. Again, the larger the value for ε_3 chosen by the third vehicle, the smaller the result for γ_3.
6) The third vehicle passes its $\{\varepsilon_3, \gamma_3\}$ to the fourth vehicle.
7) Vehicles 4-N_v repeat steps (4)–(6) until (35) cannot be satisfied realistically. This will determine the size of the platoon at formation. As noted, this size will depend on the

TABLE II
EXPECTED PLATOON SIZES FOR VARYING PLATOON COMPOSITIONS AND CONTROL FLEXIBILITY

Platoon Constitution			N_v
Vehicle Types	Move Suppression[a]	Leader Move Suppression	range
similar	tight	moderate-tight	20-115
similar	tight	loose	10-60
similar	moderate	tight	3-4
similar	moderate	loose-moderate	2-3
heterogeneous	moderate-tight	tight	6-8
heterogeneous	moderate-tight	loose-moderate	2-5

[a] We assume $\gamma_i > 0.01, i = 1, \ldots, N_v$. Tight versus moderate refers to smaller versus larger ε_i values, respectively, which reflect less versus more control flexibility.

order of the vehicles in the heterogeneous platoon. It will also depend on each vehicle's choice of $\{\beta_i, \varepsilon_i\}$.

Using (35) to lower bound acceptable values of γ_i and the monotonically decreasing nature of γ_i as $i \rightarrow N_v$ implies that the size of the platoon is inversely related to the size of ϵ_i (i.e., the tightness of the move suppression constraints) chosen by the vehicles. Finally, the sequential process for chosing $\{\varepsilon_i, \gamma_i\}$ is performed only at the time of platoon formation. Members of the platoon will use these values whenever there is a re-initialization to a new desired speed and until there is a re-configuration of the platoon (e.g., vehicles split from the platoon).

VIII. PLATOON SIZE

Given Algorithm IV.3, we now address the expected platoon size. It is assumed that (35) becomes unrealistic to satisfy when $\gamma_{N_v+1} < 0.01$. Table II summarizes a qualitative analysis based on the platoon formation process in Section VI with $\xi_i = \varepsilon_i$. From Table II, the tradeoff between move suppression flexibility and platoon size is apparent. From this tradeoff, the following recommendations are presented.

1) Efforts should be made to choose a leader with small ε_1 and the ability to incorporate road geometry into its computed profiles. The whole platoon will benefit from fuel-efficient profiles from this type of leader.
2) For platoons focused on fuel-efficiency through drafting, long platoons are possible with tight move suppression. Truck platoons fall within this category where drafting provides more benefit than having each truck individually optimize its profile within the move suppression bounds for fuel savings.
3) For platoons focused on comfort (e.g., cooperative adaptive cruise control), move suppression should be relaxed to allow the vehicle to harness whatever eco-optimized benefit is possible. This type of platoon assumes that greater fuel efficiency is possible from modifying individual vehicle profiles within the move suppression bounds, than by tightly drafting behind other vehicles.

IX. CONCLUSIONS

In this paper, a distributed implementation of receding horizon control is formulated to address vehicle platooning. Generality of the approach allows the subsystem dynamics to be nonlinear and heterogeneous, which is seen as a considerable benefit from an implementation point of view. Receding horizon control also affords the ability to handle other constraints on the states and inputs. In our distributed implementation, updates occur in parallel within a common global clock period, but tasks within the period need not be synchronized. Moreover, neither computation nor communication is assumed to happen instantaneously. Stability of the approach is achieved through parametric conditions on the weights in the cost function of each local optimal control problem. By adding a move-suppression constraint in each local optimal control problem, the stabilizing controller becomes string stabilizing. We also note that our string-stable controller does not require acceleration information from the lead car, nor does it require any continuous information from the lead car to cars down the string. This is in stark contrast with string-stable controllers that fundamentally require lead car acceleration in addition to position error and speed [22]. Procedures for platoon formation that ensure predecessor-follower string stability was described, and indicate that platoons between tens and hundreds of vehicles are possible. The size of a platoon is dependent upon the individual choices made by the constituent vehicles. Larger platoons are possible when vehicles behave homogeneously with tightly constrained paths, a trend consistent with the fact that methods derived for spatially invariant (homogeneous) systems are applicable to a countably infinite number of vehicles [10], [21].

REFERENCES

[1] L. Acar, "Boundaries of the receding horizon control for interconnected systems," *J. Optim. Theory Appl.*, vol. 84, no. 2, 1995.

[2] J. T. Betts*, Practical Methods for Optimal Control Using Nonlinear Programming*. Philadelphia, PA: SIAM, 2001.

[3] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Syst. Mag.*, Feb. 2002.

[4] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive scheme with guaranteed stability," *Automatica*, vol. 14, no. 10, pp. 1205–1217, 1998.

[5] W. B. Dunbar, "Distributed receding horizon control of multiagent systems," Ph.D. thesis, California Inst. Technol., Pasadena, CA, 2004.

[6] W. B. Dunbar, "Distributed receding horizon control of dynamically coupled nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 7, pp. 1249–1263, Jul. 2007.

[7] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[8] P. Falcone, F. Borrelli, E. H. Tseng, J. Asgari, and H. Davor, "Low complexity MPC schemes for integrated vehicle dynamics control problems," in *Proc. 9th Int. Symp. Advanced Vehicle Control (AVEC '08)*, 2008, pp. 875–880.

[9] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 324–338, Jan. 2008.

[10] M. R. Jovanovic and B. Bamieh, "On the ill-posedness of certain vehicular platoon control problems," *IEEE Trans. Autom. Control*, vol. 50, no. 9, pp. 4583–4588, Sep. 2005.

[11] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, Dec. 2006.

[12] N. J. Kohut, J. K. Hedrick, and F. Borrelli, "Integrating traffic data and model predictive control to improve fuel economy," in *Proc. 12th IFAC Symp. Control in Transportation Systems*, 2009, pp. 2806–2813.

[13] Y. Kuwata, A. G. Richards, T. Schouwenaars, and J. P. How, "Distributed robust receding horizon control for multi-vehicle guidance," *IEEE Trans. Control Syst. Technol.*, vol. 15, no. 4, Jul. 2007.

[14] J. Liu, D. M. de la Pena, and P. D. Christofides, "Distributed model predictive control of nonlinear systems subject to asynchronous and delayed measurements," *Automatica*, vol. 46, no. 1, pp. 52–61, 2010.

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

[16] H. Michalska and D. Q. Mayne, "Robust receeding horizon control of contrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 38, pp. 1623–1632, 1993.

[17] N. Motee and B. Sayyar-Rodsari, "Optimal partitioning in distributed model predictive control," in *Proc. IEEE Amer. Control Conf.*, 2003, pp. 5300–5305.

[18] A. Richards and J. How, "Robust model predictive control with imperfect information," in *Proc. IEEE Amer. Control Conf.*, 2005, pp. 268–273.

[19] B. Saerens, M. Diehl, J. Swevers, and E. Van den Bulck, "Model predictive control of automotive powertrains—First experimental results," in *Proc. 47th IEEE Conf. Decision and Control*, 2008, pp. 5692–5697.

[20] J. D. Schwartz and M. B. Milam, "On-line path planning for an autonomous vehicle in an obstacle filled environment," in *Proc. IEEE Conf. Decision and Control*, 2008, pp. 2806–2813.

[21] J. S. Shamma and G. Arslan, "A decomposition approach to distributed control of spatially invariant systems," *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 701–707, Apr. 2006.

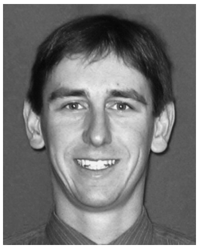
[22] D. Swaroop and J. K. Hedrick, "String stability of interconnected systems," *IEEE Trans. Autom. Control*, vol. 41, no. 3, pp. 349–357, Mar. 1996.

[23] A. N. Venkat, J. B. Rawlings, and S. J. Wright, "Stability and optimality of distributed model predictive control," in *Proc. IEEE Conf. Decision and Control/IEE Eur. Control Conf.*, Seville, Spain, 2005.

[24] E. T. Whittaker and G. N. Watson*, A Course in Modern Analysis*, 4th ed. Cambridge, U.K.: Cambridge Univ. Press, 1990.

William B. Dunbar (S'03–M'04) received the B.S. degree in engineering science and mechanics from the Virginia Polytechnic Institute and State University, Blacksburg, in 1997, the M.S. degree in applied mechanics and engineering science from the University of California, San Diego, in 1999, and the Ph.D. in control and dynamical systems from the California Institute of Technology, Pasadena, in 2004.

He is currently an Associate Professor in the Department of Computer Engineering, University of California, Santa Cruz. His research is focused on applications of feedback control to problems in biophysics and sequencing with nanopores.



Derek S. Caveney (S'00–M'05) received the B.Sc.E. degree in applied mathematics from Queen's University, Kingston, ON, Canada, in 1999, and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of California, Berkeley, in 2001 and 2004, respectively.

He is currently a Principal Scientist with the Toyota Technical Center, Toyota Motor Engineering & Manufacturing North America, Ann Arbor, MI. From 2004 until 2005, he was a Visiting Postdoctoral Scholar with the Center for Collaborative Control of Unmanned Vehicles, Berkeley, CA. His interests include advanced vehicle control systems for safety, ecology, and mobility.