2017 American Control Conference
Sheraton Seattle Hotel
May 24–26, 2017, Seattle, USA

Decentralized Control with Moving-Horizon Linear Switched Systems: Synthesis and Testbed Implementation

Joao P. Jansch-Porto and Geir E Dullerud[1]
Department of Mechanical Science and Engineering
University of Illinois, Urbana, Illinois - 61801
Email: {janschp2,dullerud}@illinois.edu

***Abstract*— In this paper, we improve recent results on the decentralized switched control problem to include the moving horizon case and apply it to a testbed system. Using known derivations for a centralized controller with look-ahead, we were able to extend the decentralized problem with finite memory to include receding horizon modal information. We then compare the performance of a switched controller with finite memory and look-ahead horizon to that of a linear time independent (LTI) controller using a simulation. The decentralized controller is further tested with a real-world system comprised of multiple model-sized hovercrafts.**

I. Introduction

In this paper, we are interested in the application of decentralized control to nested systems with switching dynamics. Nested systems represent a hierarchy of subsystems with a unidirectional flow of information amongst them, such as the configuration presented in Fig. 1. This setup can be encountered in many real-world applications, such as economic theory, power systems, and vehicle formations.

For the controller synthesis, we will consider the H_∞-type cost criteria. This criteria is also referred to as disturbance attenuation or root-mean square gain. The main idea behind this performance measure is to minimize the effect of the worst-case disturbance onto the energy of the system.

Explicit state-space solutions to H_∞ optimal control of nested systems has only recently been developed. rior work on distributed systems using state-space techniques include [1] and [2], which provide sufficient conditions for controller existence and then explicit construction conditionally. In [3], the author considered the decentralized control of continuous-time time-invariant systems with nested interconnection structure, presenting tight conditions for controller synthesis. The discrete-time time-varying version of the H_∞ optimal control problem of interconnected systems was considered in [4]. Other authors have discussed the decentralized control of nested systems in the past ([5] and [6]), however considering different performance criteria.

For our application, we have to consider a nested discrete time switching system. Synthesis conditions for a finite-path dependent centralized controller were given in [7]. The results in [7] were further extended in [8] to allow the controller access to a finite number of future parameters. The synthesis results for a finite-path controller for nested systems was presented in [9]. In this paper we present an extension to [9] considering receding horizon modal information, as it was shown for the centralized case in [8].

Since we are interested in seeing how the switched controller performs in a real-world system, we have implemented the decentralized controller to multiple robots. We utilize testbed comprised of the HoTDeC hovercrafts developed at the University of Illinois. This testbed was previously used in [10] for a persuit-evasion problem. In this paper we consider a vehicle formation scenario, where a leader follows a desired reference signals, and the following vehicles trail a certain distance behind. We analized the performance of the system both in simulation and experimentation.

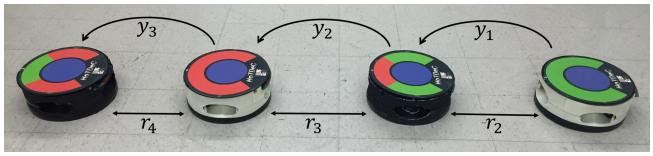

Fig. 1: Hovercraft Experimental setup. The rightmost vehicle follows a reference signal, while each other vehicle maintains a distance r_i away from the robot directly in front.

II. Switching Systems

A. Notation

We denote the space of n-dimensional symmetric, positive-definite, and positive-semidefinite matrices by $\mathbb{S}^n$, $\mathbb{S}^n_+$, and $\bar{\mathbb{S}}^n_+$. For any matrix W, $W_\perp$ denotes full column rank matrices satisfying $\mathrm{Im}(W_\perp) = \ker(W)$, $W_\perp^\top W_\perp = I$. We let ℓ^n be the space of infinite indexed sequence of elements $x = (x(0), x(1), x(2), \dots)$ with $x(t) \in \mathbb{R}^n$ for $t \in \mathbb{N}_0$. A subspace of ℓ^n is the Hilbert space ℓ_2^n (or simply ℓ_2), with norm $\|x\| = \sum_{t=0}^{\infty} |x(t)|_2^2 < \infty$.

To denote the number of decentralized subsytstems in the nested setup, we use M. The space of block-lower triangular matrices takes the form

$$\begin{bmatrix} H_{11} & 0 & \dots & 0 \\ H_{21} & H_{22} & & 0 \\ \vdots & & \ddots & \vdots \\ H_{M1} & H_{M2} & \dots & H_{MM} \end{bmatrix}$$

by $\mathcal{S}\big((m_1, \dots, m_M), (k_1, \dots, k_M)\big)$ so that $H_{ij} \in \mathbb{R}^{m_i \times k_j}$ and $H_{ij} = 0$ for $i < j$. Additionally, the notation $\mathcal{J} = \{1, \dots, M\}$ and $\bar{\mathcal{J}} = \{0, \dots, M\}$ is utilized. For partitioned

[1]The authors were partially supported by AFOSR Grant FA9550-15-1-0059, and NSF Grant 1629949.

symmetric matrices, say $\begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3 \end{bmatrix}$, we will suppress repeated sub-blocks as $\begin{bmatrix} X_1 & X_2 \\ \cdot & X_3 \end{bmatrix}$. We will also write inequalities of the form $W^\top G W \succ 0$ as $[\bullet]^\top G W \succ 0$, for some matrix G.

B. Preliminaries

1) Mode Dependent Switched Systems: A switched system is defined to be a multi-model system that allows transitions among operation models, where each mode corresponds to a distinct state-space model ([11]). The system dynamics are given by

$$\begin{aligned} x(t+1) &= A_{\theta(t)}x(t) + B_{\theta(t)}w(t) \\ z(t) &= C_{\theta(t)}x(t) + D_{\theta(t)}w(t) \end{aligned} \tag{1}$$

where the system matrices depend on the switching signal $\theta(t)$. We assume that our switching signal takes values from a discrete and finite set $\mathcal{N} = \{1, \dots, n_s\}$, and that switching between values in time is governed by a finite-state automata. The set of *admissible* sequences of length $r \in \mathbb{N}_0$ generated by such an automaton is denoted as $\mathcal{A}_r$.

For the decentralized control problem, we consider the following mode-dependent switched plant:

$$\begin{aligned} x(t+1) &= A_{\theta(t)}x(t) + B^w_{\theta(t)}w(t) + B^u_{\theta(t)}u(t) \\ z(t) &= C^z_{\theta(t)}x(t) + D^{zw}_{\theta(t)}w(t) + D^{zu}_{\theta(t)}u(t) \\ y(t) &= C^y_{\theta(t)}x(t) + D^{yw}_{\theta(t)}w(t) \end{aligned} \tag{2}$$

Here $w(t) \in \mathbb{R}^{n^w}$ is the disturbance input, $z(t) \in \mathbb{R}^{n^z}$ is the performance output, $u(t) \in \mathbb{R}^{n^u}$ is the control input, and $y(t) \in \mathbb{R}^{n^y}$ is the measurement available to the controller. The states, inputs, and outputs are partitioned as

$$x(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_M(t) \end{bmatrix},\ u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_M(t) \end{bmatrix},\ y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_M(t) \end{bmatrix}$$

where $x_i(t) \in \mathbb{R}^{n_i}$, $u_i(t) \in \mathbb{R}^{n^u_i}$, and $y_i(t) \in \mathbb{R}^{n^y_i}$. The dimensions satisfy $n = \sum_{i=1}^M n_i$, $n^u = \sum_{i=1}^M n^u_i$, and $n^y = \sum_{i=1}^M n^y_i$. We also introduce the tuple $\bar{n} = (n_1, \dots, n_M)$ and similarly define $\bar{n}^u$ and $\bar{n}^y$.

Since we are interested in nested systems with unidirectional flow of information, as in Fig. 1, we want to enforce a certain structure to our system matrices. To this end, we make the following assumption:

Assumption 1. We assume that $A_\phi \in \mathcal{S}(\bar{n}, \bar{n})$, $B^u_\phi \in \mathcal{S}(\bar{n}, \bar{n}^u)$, and $C^y_\phi \in \mathcal{S}(\bar{n}^y, \bar{n})$ for all $\phi \in \mathcal{N}$.

This assumption restricts our controller synthesis problem to decentralized systems.

2) Path Dependent Systems: Now consider the switched system

$$\begin{aligned} x(t+1) &= A_{\Omega(t)}x(t) + B_{\Omega(t)}w(t) \\ z(t) &= C_{\Omega(t)}x(t) + D_{\Omega(t)}w(t) \end{aligned} \tag{3}$$

whose system matrices at time t depend on a switching path $\Omega(t) = (\theta(t-L), \dots, \theta(t)) \in \mathcal{A}_{L+1}$ consisting of $L+1$ recent values of the switching parameters. We refer to these types of systems as *finite-path dependent* systems

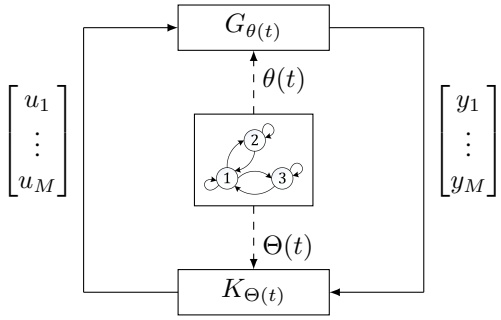

Fig. 2: Interconnection diagram of a controller, $K_{\Theta(t)}$, with plant, $G_{\theta(t)}$, where $\Theta(t)$ is the switching sequence.

with *memory* of length L. We can modify such systems to be mode-dependent by introducing induced automata to reflect the path dependence. This is done by assuming the induced automata state-space to be $\tilde{\mathcal{N}} = \mathcal{A}_{L+r}$, for admissible sequences of length $r > 0$.

For $r > 0$ we define the sequences $\bar{\Phi}, \underline{\Phi} \in \mathcal{A}_{r+L}$, $\Phi_\dagger \in \tilde{\mathcal{N}} = \mathcal{A}_{L+1}$, and $\Phi_\star \in \mathcal{N}$ as

$$\begin{aligned} \bar{\Phi} &\coloneqq (\beta_1, \dots, \beta_{r+L}), \quad \underline{\Phi} \coloneqq (\beta_0, \dots, \beta_{r+L-1}), \\ \Phi_\dagger &\coloneqq (\beta_r, \dots, \beta_{r+L}), \quad \Phi_\star \coloneqq \beta_{r+L}. \end{aligned}$$

3) Switching control: For the plant (2), our goal is to design a finite-dimensional, finite-path dependent linear controller with block lower triangular sparsity structure. We use the following state space representation for our controller

$$\begin{aligned} x^K(t+1) &= A^K_{\Omega(t)}x^K(t) + B^K_{\Omega(t)}y(t) \\ u(t) &= C^K_{\Omega(t)}x^K(t) + D^K_{\Omega(t)}y(t) \end{aligned} \tag{4}$$

For a controller with memory L, the switching path corresponds to $\Omega(t) = (\theta(t-L), \dots, \theta(t)) \in \mathcal{A}_{L+1}$. The controller state $x^K(t) \in \mathbb{R}^{n^K}$ is partitioned as $\left[(x^K_1(t))^\top \dots (x^K_M(t))^\top\right]$ with $x^K_i(t) \in \mathbb{R}^{n^K_i}$, thus satisfying $n^K = n^K_1 + \dots + n^K_M$. Our goal is then to design the above controller with the following associated structured controller matrices for every admissible sequence $\Psi \in \mathcal{A}_{L+1}$,

$$\begin{aligned} A^K_\Psi &\in \mathcal{S}(\bar{n}^K, \bar{n}^K), \qquad B^K_\Psi \in \mathcal{S}(\bar{n}^K, \bar{n}^y) \\ C^K_\Psi &\in \mathcal{S}(\bar{n}^u, \bar{n}^K), \qquad D^K_\Psi \in \mathcal{S}(\bar{n}^u, \bar{n}^y) \end{aligned} \tag{5}$$

where $\bar{n}^K = (n^K_1, \dots, n^K_M)$. Thus, the resulting controller has a y to u mapping with a lower triangular sparsity structure.

We can define the following kernel space matrices for $i \in \bar{\mathcal{J}}$, $\phi \in \mathcal{N}$,

$$\begin{aligned} N^y_{i,\phi} &= \begin{bmatrix} N^{y,x}_{i,\phi} \\ N^{y,w}_{i,\phi} \end{bmatrix} = \begin{bmatrix} (E^y_i)^\top C^y_\phi & (E^y_i)^\top D^{yw}_\phi \end{bmatrix}_\perp \\ N^u_{i,\phi} &= \begin{bmatrix} N^{u,x}_{i,\phi} \\ N^{u,z}_{i,\phi} \end{bmatrix} = \begin{bmatrix} (\bar{E}^u_i)^\top (B^u_\phi)^\top & (\bar{E}^u_i)^\top (D^{zu}_\phi)^\top \end{bmatrix}_\perp \end{aligned}$$

Additionally, consider the following construction which will be useful for the controller synthesis,

$$N_{i,\phi} = \begin{bmatrix} N^u_{i,\phi} & 0 \\ 0 & N^y_{i,\phi} \end{bmatrix} \tag{6}$$

The closed loop scaling matrices are denoted by $X_\Psi^C \in \mathbb{S}_+^{n+n^K}$, defined for each $\Psi \in \mathcal{A}_L$. These matrices are partitioned into plant and controller sections as

$$X_\Psi^C = \begin{bmatrix} X_\Psi & X_\Psi^{GK} \\ (X_\Psi^{GK})^\top & X_\Psi^K \end{bmatrix}, (X_\Psi^C)^{-1} = \begin{bmatrix} Y_\Psi & Y_\Psi^{GK} \\ (Y_\Psi^{GK})^\top & Y_\Psi^K \end{bmatrix} \tag{7}$$

with $X_\Psi, Y_\Psi \in \mathbb{S}_+^n$, $X_\Psi^{GK}, Y_\Psi^{GK} \in \mathbb{R}^{n\times n^K}$, and $X_\Psi^K, Y_\Psi^K \in \mathbb{S}_+^{n^K}$. We further define the following for $i \in \bar{\mathcal{J}}$,

$$\begin{aligned} Z_{i,\Psi} &:= \left\{ X_\Psi - X_\Psi^{GK} \bar{E}_i^K \big((\bar{E}_i^K)^\top X_\Psi^K \bar{E}_i^K \big)^{-1} (X_i^{GK} \bar{E}_i^K)^\top \right\}^{-1} \\ &= Y_\Psi - Y_\Psi^{GK} E_i^K \big((E_i^K)^\top Y_\Psi^K E_i^K \big)^{-1} (Y_\psi^{GK} E_i^K)^\top \end{aligned} \tag{8}$$

We then define the following associated matrices

$$\begin{aligned} &Z_{i,\Psi}^a := (E_i^\top Z_{i,\psi} E_i)^{-1},\ Z_{i,\Psi}^b := -Z_{i,\psi}^a (E_i^\top Z_{i,\psi} \bar{E}_i) \\ &Z_{i,\Psi}^c := \bar{E}_i^\top Z_{i,\psi} \bar{E}_i - (E_i Z_{i,\psi} \bar{E}_i)^\top (E_i^\top Z_{i,\psi} E_i)^{-1} E_i^\top Z_{i,\psi} \bar{E}_i \end{aligned} \tag{9}$$

The matrices above give us the following factorization

$$Z_{i,\psi} = Z_{i,\psi}^l (Z_{i,\psi}^u)^{-1} = (Z_{i,\psi}^u)^{-\top} (Z_{i,\psi}^l)^\top \tag{10}$$

with

$$Z_{i,\psi}^l = \begin{bmatrix} I & 0 \\ -(Z_{i,\psi}^b)^\top & Z_{i,\psi}^c \end{bmatrix}, \quad Z_{i,\psi}^u = \begin{bmatrix} Z_{i,\psi}^a & Z_{i,\psi}^b \\ 0 & I \end{bmatrix}. \tag{11}$$

We are now ready to state the main result presented in [9]:

Theorem 1. Consider the mode-dependent systems (2) along with Assumption 1. There exists a synthesis of a finite-path dependent controller (4) which

(i) is structured as (5)
(ii) has dimensions $\{n_i^K\}_{i=1}^M$
(iii) achieves closed loop performance $\|w \mapsto z\| < 1$

if and only if there exists an $L \in \mathbb{N}_0$ and matrices $\{Z_{i,\Psi}^a, Z_{i,\Psi}^b, Z_{i,\Psi}^c\}_{i\in\bar{\mathcal{J}}, \Psi\in\mathcal{A}_L}$ satisfying the following

$$Z_{i,\Psi}^a \succ 0,\ Z_{i,\Psi}^c \succ 0 \text{ for all } i \in \bar{\mathcal{J}},\ \Psi \in \mathcal{A}_L \tag{12a}$$

$$\left[\bullet\right]^\top \left[\begin{array}{cc|c} (Z_{i,\bar{\Phi}}^u)^\top Z_{i,\bar{\Phi}}^l & 0 & (Z_{i,\bar{\Phi}}^u)^\top A_{\Phi_\star} Z_{i,\Phi}^l \ (Z_{i,\bar{\Phi}}^u)^\top B_{\Phi_\star}^w \\ 0 & I & C_{\Phi_\star}^z Z_{i,\Phi}^l \quad D_{\Phi_\star}^{zw} \\ \hline \cdot & \cdot & (Z_{i,\Phi}^u)^\top Z_{i,\Phi}^l \quad 0 \\ \cdot & \cdot & 0 \quad I \end{array}\right] N_{i,\Phi_*} \succ 0$$

$$\text{for all } i \in \bar{\mathcal{J}},\ \Phi \in \mathcal{A}_{L+1}, \tag{12b}$$

$$\begin{bmatrix} (Z_{i,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i,\Psi}^l)^\top Z_{i-1,\Psi}^u \\ (Z_{i-1,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i-1,\Psi}^u)^\top Z_{i-1,\Psi}^l \end{bmatrix} \succeq 0 \tag{12c}$$

$$\operatorname{rank} \begin{bmatrix} (Z_{i,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i,\Psi}^l)^\top Z_{i-1,\Psi}^u \\ (Z_{i-1,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i-1,\Psi}^u)^\top Z_{i-1,\Psi}^l \end{bmatrix} \le n + n_i^K \tag{12d}$$

Remark 1. If a closed loop performance of $\|w \mapsto z\| < \gamma$ is desired, Theorem 1 can be updated to have C_ϕ^z, C_ϕ^y, D_ϕ^{zw}, D_ϕ^{zu}, and D_ϕ^{yw} scaled by γ^{-1} for all $\phi \in \mathcal{N}$. The controller obtained for this modified system using the procedure above can be used to find the desired controller by scaling B_Ψ^K and D_Ψ^K with γ^{-1} for all Ψ.

C. Receding Horizon

Now we want to extend the previous result so that our nested switching system also accounts for future modes, as presented for the centralized case in [8]. For this end, we define $\Phi^+, \Phi^- \in \mathcal{A}_{r+L+H}$, $\Phi_\dagger \in \mathcal{A}_{r+L+H}$, and $\Phi_0 \in \mathcal{N}$ as

$$\begin{aligned} &\Phi^+ := (\beta_1, \dots, \beta_{r+L+H}), \quad \Phi^- := (\beta_0, \dots, \beta_{r+L+H-1}), \\ &\Phi_\dagger := (\beta_0, \dots, \beta_{r+L+H}), \quad \Phi_0 := \beta_{r+L}. \end{aligned}$$

Note that the above definitions are equivalent to the case without any look-ahead horizon.

Now, consider the switched system

$$\begin{aligned} x(t+1) &= A_{\Theta(t)} x(t) + B_{\Theta(t)} w(t) \\ z(t) &= C_{\Theta(t)} x(t) + D_{\Theta(t)} w(t) \end{aligned} \tag{13}$$

For a controller with memory L and look-ahead horizon H, the above system matrices at time t depend on a switching path given by $\Theta(t) = \big(\theta(t-L), \dots, \theta(t), \dots, \theta(t+H)\big) \in \mathcal{A}_{L+H+1}$.

We can modify Lemma 4 in [9] to include a look-ahead horizon, as it was shown in [8].

Lemma 1. The finite-path dependent system (13) with memory $L \in \mathbb{N}_0$ and look-ahead horizon $H \in \mathbb{N}_0$ is exponentially stable and satisfies $\|w \mapsto z\| < 1$ if and only if there exists an $r \in \mathbb{N}_0$ and a set of positive-definite matrices $\{X_\Psi\}_{\Psi\in\mathcal{A}_{r+L+H}}$ satisfying

$$\begin{bmatrix} X_{\Phi^-} & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} A_{\Phi_\dagger} & B_{\Phi_\dagger} \\ C_{\Phi_\dagger} & D_{\Phi_\dagger} \end{bmatrix}^\top \begin{bmatrix} X_{\Phi^+} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{\Phi_\dagger} & B_{\Phi_\dagger} \\ C_{\Phi_\dagger} & D_{\Phi_\dagger} \end{bmatrix} \succ 0$$

for all $\Phi \in \mathcal{A}_{r+L+H+1}$.

Now we can retrace the steps in [9] to obtain a set of LMIs that give necessary and sufficient conditions for the existence of the path dependent controller.

Theorem 2. Consider the mode-dependent systems (2) along with Assumption 1. There exists a synthesis of a finite-path dependent controller (4) which

(i) is structured as (5)
(ii) has dimensions $\{n_i^K\}_{i=1}^M$
(iii) achieves closed loop performance $\|w \mapsto z\| < 1$

if and only if there exists $L, H \in \mathbb{N}_0$ and matrices $\{Z_{i,\Psi}^a, Z_{i,\Psi}^b, Z_{i,\Psi}^c\}_{i\in\bar{\mathcal{J}}, \Psi\in\mathcal{A}_{L+H}}$ satisfying the following

$$Z_{i,\Psi}^a \succ 0,\ Z_{i,\Psi}^c \succ 0 \text{ for all } i \in \bar{\mathcal{J}},\ \Psi \in \mathcal{A}_{L+H} \tag{14a}$$

$$\left[\bullet\right]^\top \left[\begin{array}{cc|c} (Z_{i,\Phi^+}^u)^\top Z_{i,\Phi^+}^l & 0 & (Z_{i,\Phi^+}^u)^\top A_{\Phi_0} Z_{i,\Phi^-}^l \ (Z_{i,\Phi^+}^u)^\top B_{\Phi_0}^w \\ 0 & I & C_{\Phi_0}^z Z_{i,\Phi^-}^l \quad D_{\Phi_0}^{zw} \\ \hline \cdot & \cdot & (Z_{i,\Phi^-}^u)^\top Z_{i,\Phi^-}^l \quad 0 \\ \cdot & \cdot & 0 \quad I \end{array}\right] N_{i,\Phi_*} \succ 0$$

$$\text{for all } i \in \bar{\mathcal{J}},\ \Phi \in \mathcal{A}_{L+H+1}, \tag{14b}$$

$$\begin{bmatrix} (Z_{i,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i,\Psi}^l)^\top Z_{i-1,\Psi}^u \\ (Z_{i-1,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i-1,\Psi}^u)^\top Z_{i-1,\Psi}^l \end{bmatrix} \succeq 0 \tag{14c}$$

$$\operatorname{rank} \begin{bmatrix} (Z_{i,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i,\Psi}^l)^\top Z_{i-1,\Psi}^u \\ (Z_{i-1,\Psi}^u)^\top Z_{i,\Psi}^l & (Z_{i-1,\Psi}^u)^\top Z_{i-1,\Psi}^l \end{bmatrix} \le n + n_i^K \tag{14d}$$

Remark 2. Note that the solution to the above inequalities depend both on the switching path and the current mode of the system. Hence the controller matrices will change based on the choice of L and H.

Remark 3. If one desires to obtain a decentralized controller

for a linear time invariant (LTI) plant, the results above can still be used by setting $\mathcal{N} = \{1\}$. That is, the automaton only generates one admissible sequence. As a result, for any L and H, there exists only one sequence in $\mathcal{A}_{L+H+1}$, and we can then drop the subscripts in Theorem 2.

For controller construction, refer to Theorem 16 in [9].

D. Example

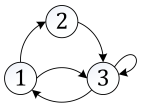

Fig. 3: Switching automata used in the Example.

Let us consider the example shown in [9] in order to demosntrate the benefits of the look-ahead controller. The two player system switches between 3 different modes, with a switching sequence generated by the automaton in Fig. 3. The corresponding system matrices are chosen as

$$A_1 = A_2 = \begin{bmatrix} 1.4 & 0 \\ 0.2 & 1.4 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0.7 & 0 \\ 0.2 & 0.7 \end{bmatrix}$$
$$B_1^u = B_2^u = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad B_3^u = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$
$$C_1^y = C_2^y = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad C_3^y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$D_1^{zu} = D_2^{zu} = [0\ 1], \quad D_3^{zu} = [4\ 0]$$

and the following defined for $\phi \in \{1, 2, 3\}$

$$B_\phi^w = \begin{bmatrix} 1 \\ 1 \end{bmatrix},\ D_\phi^{yw} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ C_\phi^z = [0.5\ 2],\ D_\phi^{zw} = 0.5$$

Here we have chosen dimensions $n_1 = n_2 = n_1^u = n_2^u = n_1^y = n_2^y = n^z = n^w = 1$.

For different memory lengths, the above system was examined with $n_i^K = 2$ for $i \in \mathcal{J}$. The optimal bound $\|w \mapsto z\| < \gamma$ was found using a bisection algorithm, and the values for different memory lengths are tabulated below.

TABLE I: Optimal performance bound for the controller.

L/H	0	1	2
0	∞	5.467	3.000
1	5.468	3.160	
2	3.663		
3	3.606		

For $H = L = 0$, the system is not stabilizable, resulting in infinite norms. As one would intuitively expect, having a preview of future modes gives much better H_∞ bounds than just relying on previous modes.

III. Hardware

The hardware consists of the HoTDeC (Hovercraft Testbed for Decentralized Control) vehicle, which was developed at the University of Illinois at Urbana-Champaign, and a Vision system. Detailed information about the HoTDeC and earlier versions of the system can be found in [12] and [13].

A. HoTDeC

There are two different types of HoTDeC bodies: 3D printed and precision machined. Each vehicle has five thrusters, each generating up to 1.2 N of thrust: one generates lift, and the others generate forces in the x- and y-direction. Due to their positioning, we can also use a combination of them to generate a moment and rotate the hovercraft body.

The HoTDeC consists of various electronic boards. The Powerboard regulates the voltage and power distribution for the hovercraft. The Main board is in charge of the integration of peripherals such the Gumstix and the Digital Signal Processor (DSP). The Gumstix is the main processor of each vehicle, running on a real-time embedded Linux OS, and the DSP used is a Texas Instruments TMS320F28335. Each motor board includes a hall effect sensor, used to measure the thruster angular velocity, and an H-bridge.

B. Dynamical Model

The dynamic model of the HoTDeC is defined by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Gw(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned} \tag{15}$$

where the $x(t)$ states are the positions and velocities in Cartesian coordinates, with the following matrices:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\beta_x}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\beta_y}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{\beta_\theta}{m} \end{bmatrix},\ B = G = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Where $\beta_{trans} = 3.5 \times 10^{-3} N \cdot s/m$ is the coefficient of translational friction, and $\beta_\theta = 2.63 \times 10^{-4} N \cdot m \cdot s/rad$ is the coefficient of rotational friction. Detailed information about the actuator disturbance, $w(t)$, and the measurement noise, $v(t)$, are presented in [14].

TABLE II: Mechanical parameters of the HoTDeC.

Body Type	Mass, m (kg)	M. of Inertia, J $(kg \cdot m^2)$
Styrofoam	2.312	0.037
3D Printed	2.642	0.043

The DSP board runs 5 independent proportional-integral (PI) loops to control the speed of each thruster. We assume that the relationship between force F and thruster angular velocity ω is $\omega^2 = 2560000 \times F$. Each thruster is modeled as

$$\begin{aligned} \dot{x} &= -8.6x + u \\ w^2 &= 20868760x \end{aligned} \tag{16}$$

with

$$F = \begin{cases} 8.1518x - 0.02 & \text{if } x > 0.00246 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

C. Vision System & Network

To determine the position of the hovercrafts, the HoTDeC lab has a setup of six cameras covering a total area of approximately 3m by 2m. One computer processes the information from each camera, while another computer merges the information to remove any duplicate data from the overlapping areas and broadcasts the data. The Vision system differentiates between different hovercrafts and determines angle of each by using the patterns on their tops, as seen in Fig. 1. We use the ZeroMQ messaging system to communicate with the vehicles and the Vision system.

IV. Controller Design

A. Controller

Our goal is to have the hovercrafts follow each other in a platoon of vehicles, as with the system depicted by Fig. 1. Each vehicle will try to keep a set distance from the one in front of it, with the first vehicle following a user defined path. First, to set up our controller, we have to generate the state space system which describes the nested loop.

Let $\varepsilon_i \in \mathbb{R}^{n_i}$ be the measured position vector of the i-th robot that is available to its followers. Here we assume that we get full state information via an estimator. Then, if we desire that each hovercraft maintain a distance $r_i \in \mathbb{R}^{n_i}$ away from its leader, we can choose the performance output $e_i = y_i - \varepsilon_{i-1} - r_i$, where that of the first hovercraft is $e_1 = y_1 - r_1$.

Synthesis of the decentralized switching controller will be based on the un-weighted generalized plant of the nested structure

$$\begin{bmatrix} z_e \\ z_u \\ y \end{bmatrix} = \left[\begin{array}{c|cc} A & B^w & B^u \\ \hline C^z & D^{zw} & D^{zu} \\ C^y & D^{yw} & 0 \end{array}\right] \begin{bmatrix} d \\ u \end{bmatrix} \tag{18}$$

where $z_e = \begin{bmatrix} e_1, \ldots, e_M \end{bmatrix}^\top$ is the position performance output, $z_u = \begin{bmatrix} u_1, \ldots, u_M \end{bmatrix}^\top$ is the controller performance output, $y = \begin{bmatrix} y_1, \ldots, y_M \end{bmatrix}^\top$ is the measurement output, $d = \begin{bmatrix} r_1, \ldots, r_M \end{bmatrix}^\top$ is the disturbance input, and $u = \begin{bmatrix} u_1, \ldots, u_M \end{bmatrix}^\top$ is the controller input. The dynamical model was discretized with a sampling time of $Ts = 0.1$ seconds.

For our experiment, the hovercraft will operate near obstacles in either the x- or y-direction. To avoid any collisions with the environment, the performance output switches between three modes given by

$$\begin{aligned} z_1 &= \begin{bmatrix} 4x & 2\dot{x} & 4y & 2\dot{y} & 5\theta & 2\dot{\theta} & 0.5u_x & 0.5u_y & 0.6u_\theta \end{bmatrix} \\ z_2 &= \begin{bmatrix} 6x & 2\dot{x} & 3y & 2\dot{y} & 5\theta & 2\dot{\theta} & 0.5u_x & 0.5u_y & 0.6u_\theta \end{bmatrix} \\ z_3 &= \begin{bmatrix} 3x & 2\dot{x} & 6y & 2\dot{y} & 5\theta & 2\dot{\theta} & 0.5u_x & 0.5u_y & 0.6u_\theta \end{bmatrix} \end{aligned} \tag{19}$$

where mode 1 represents unobstructed operation, and modes 2 and 3 represent operation near an obstacle in the x- or y-direction respectively. The weights were chosen based on the limitations of the hovercrafts, so as to not saturate the system. All admissible sequences in our system can be generated by the automaton in Fig. 2. Note that the weighting vectors above are for a single hovercraft. For the nested system, these vectors will be expanded accordingly, with all hovercrafts subjected to the same weights. CVX, a MATLAB-based toolbox for convex optimization ([15]), was used to solve the LMIs in Theorem 2.

For our problem, we choose the current and future mode based on the position and velocities of the robots. If the plant switching signal is not available, the controller switching signal can be generated by numerous methods, such as the one proposed in [16].

B. Estimator

To deal with the noisy input from the Vision system, we have added a Kalman Filter using the dynamics from Section III-B running at 10Hz. While the H_∞ controller is able to estimate the system states, it tends to give very conservative estimates, so we decided to implement an Kalman filter which is widely used in robotics applications.

To generate the filter gain we used the covariance matrices,

$$\begin{aligned} Q &= \mathbb{E}\left[w_n w_n^\top\right] = \begin{bmatrix} \sigma_F^2 & 0 & 0 \\ 0 & \sigma_F^2 & 0 \\ 0 & 0 & \sigma_\tau^2 \end{bmatrix}, \\ R &= \mathbb{E}\left[v_n v_n^\top\right] = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \end{aligned} \tag{20}$$

where $\sigma_x = 2.5\,mm, \sigma_F = 0.0234\,N, \sigma_y = 2.23\,mm, \sigma_\tau = 0.0237\,N$, and $\sigma_\theta = 0.0206\,rad$.

C. Simulation

In order to quickly test different controller weights before deploying the code to the hovercrafts, we have created a Simulink simulation where we modeled the HoTDeC dynamics and the Vision system. Here, a nested structure of four vehicles was considered. The simulation accounted for the dynamics of the hovercrafts and the thrusters themselves, simulated the code running in the DSP, and modeled the Vision system including noise and delay.

We tested how well our hovercrafts followed a circular trajectory under different controller memory and horizon lengths. In all test cases the leader followed a $0.6\,m$ wide circular path at a 0.3 Hz frequency. A LTI case was generated using only the first set of controller weights in Equation 19. All controllers were generated at their optimal bounds.

For the case of the LTI controller in Fig. 4, we can see that while the first robot (far right, red dashed line) does a good job of following the reference signal (far right, blue solid line), small deviations quickly increase as they propagate down the platoon. In the controller with memory length 2 and the controller with look-ahead horizon 2, the trailing robots do a better job at following the one in front. For example, the average root mean square position error between a follower and the robot directly in front of it is approximately 8% for the LTI case, while it is reduced to approximately 5% for the other two cases. Under the look-ahead controller, the first robot is slightly worse at following the reference signal than the other cases, likely due to noise in the system affecting the predictions of future modes.

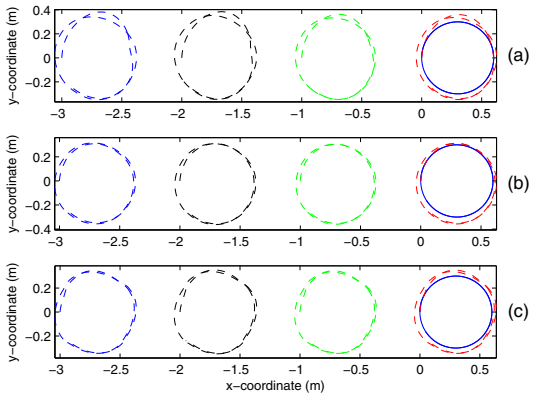

Fig. 4: Simulated paths of four hovercrafts using the decentralized controller with various switching parameters. The rightmost blue solid line represents the reference signal. Test cases: (a) LTI, (b) L=2, (c) H = 2.

V. Hardware Implementation

We tested the controller with two hovercrafts. The leader was told to move in a circle of radius $0.3\,m$, with its follower being told to trail behind at a distance of $1\,m$ in its x-coordinate position. The reference values for the trajectory were generated via sinusoidal functions with frequency of 0.3 Hz. So as to not be purely dependent upon predictions of future modes, we have tested a controller with memory $L = 1$ and look-ahead horizon $H = 1$, using the same setup as described in the previous Section.

Main results from the hovercraft experiment are presented in the plots below. It should be noted that in Fig. 5, the hovercrafts power on at the 4 second mark. They immediately follow a circular path, and the reference signal stops at 30 seconds. In both plots, the x- and y-positions are the estimated output from the Kalman Filter.

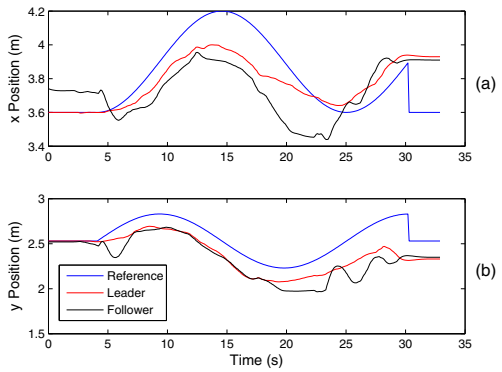

Fig. 5: Estimated position of both hovercrafts with a decentralized controller L=1, H=1. (a) x position, (b) y position.

We can see that while the leader did not follow the reference signal as well as the simulations would indicate, the follower did a very good job at keeping a constant distance behind the leader. Although the Vision System lost track of the follower at approximately the 20 second mark, the hovercraft quickly recovered to the desired distance.

VI. Conclusion

In this paper, we have presented the decentralized switched control problem and exact conditions for synthesis of a controller with finite memory. We then developed an extension of the synthesis condition for the case of a controller with receding horizon modal information. The derived controller was tested using a system of hovercrafts: first with a MATLAB simulation, and then with a real-system experiment. The simulations presented show how error can be reduced using this controller in comparison to a simply LTI system. The real hovercraft system experiment shows that the controller is functional, although there is much room for improvement in the experimental setup to better the continuity between the experimental and simulation results. Overall, the results of this paper have proven that a discrete-time linear nested system controller with finite memory and look-ahead horizon can be a useful tool.

References

[1] R. D'Andrea and G. E. Dullerud. Distributed control design for spatially interconnected systems. *IEEE Transactions on Automatic Control*, 48(9):1478–1495, Sept 2003.

[2] M. Farhood, D. Zhe, and G. E. Dullerud. Distributed control of linear time-varying systems interconnected over arbitrary graphs. *International Journal of Robust and Nonlinear Control*, 25(2):179–206, 2015.

[3] C. Scherer. Structured $\mathcal{H}_\infty$-optimal control for nested interconnections: A state-space solution. *Systems & Control Letters*, 62(12):1105 – 1113, 2013.

[4] A. Mishra, C. Langbort, and G. E. Dullerud. Decentralized control of linear time-varying nested systems with $\mathcal{H}_\infty$-type performance. In *2014 American Control Conference*, pages 5174–5179, June 2014.

[5] P. G. Voulgaris. Control of nested systems. In *American Control Conference, 2000. Proceedings of the 2000*, volume 6, pages 4442–4445 vol.6, 2000.

[6] L. Lessard and S. Lall. Optimal controller synthesis for the decentralized two-player problem with output feedback. In *2012 American Control Conference (ACC)*, pages 6314–6321, June 2012.

[7] J. W. Lee and G. E. Dullerud. Optimal disturbance attenuation for discretetime switched and markovian jump linear systems. *SIAM Journal on Control and Optimization*, 45(4):1329–1358, 2006.

[8] R. Essick, J. W. Lee, and G. E. Dullerud. Control of linear switched systems with receding horizon modal information. *IEEE Transactions on Automatic Control*, 59(9):2340–2352, Sept 2014.

[9] A. Mishra, C. Langbort, and G. E. Dullerud. Decentralized control of linear switched nested systems with ℓ_2 -induced norm performance. *IEEE Transactions on Control of Network Systems*, 2(4):420–432, Dec 2015.

[10] S. Lee, G. E. Dullerud, and E. Polak. On the real-time receding horizon control in harbor defense. In *2015 American Control Conference (ACC)*, pages 3601–3606, July 2015.

[11] D. Liberzon. *Switching in Systems and Control*. Systems & Control: Foundations & Applications. Birkhäuser Boston, 2012.

[12] S. Lee. *A Model Predictive Control Approach to a Class of Multiplayer Minmax Differential Games*. PhD thesis, University of Illinois at Urbana-Champaign, 2015.

[13] J. P. Jansch-Porto. Decentralized control of linear switched systems with receding horizon. Master's thesis, University of Illinois at Urbana-Champaign, 2016.

[14] A. Stubbs, V. Vladimerou, A. T. Fulford, D. King, J. Strick, and G. E. Dullerud. Multivehicle systems control over networks: a hovercraft testbed for networked and decentralized control. *IEEE Control Systems*, 26(3):56–69, June 2006.

[15] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[16] G. Battistelli. On stabilization of switching linear systems. *Automatica*, 49(5):1162 – 1173, 2013.