DECENTRALIZED CONTROL OF LINEAR SWITCHED SYSTEMS WITH RECEDING HORIZON

BY

JOAO PAULO JANSCH PORTO

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Professor Geir E. Dullerud

Abstract

In this thesis, we consider the decentralized switched control problem where exact conditions for controller synthesis are obtained via Linear Matrix Inequalities (LMIs). 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 MATLAB simulation. The decentralized controller is further tested with a real-world system comprised of two HoTDeC hovercrafts.

Acknowledgments

I would like to thank my adviser, Professor Geir E. Dullerud, for the guidance and support he has given me in the last two years. I also thank Seungho Lee, who helped introduced me to the HoTDeC testbed and, along with Qing Xu, gave me helpful tips on troubleshooting some problems.

I am grateful for my family support and encouragement. Last but not least, I want to thank my girlfriend, for the never ending support in my research, even when nothing seemed to go right.

Contents

Chapter 1

Introduction

1.1 Motivation

In this thesis, we are interested in the control of decentralized 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 Figure 1.1. This setup can be encountered in many real-world applications, such as economic theory, power systems, and interconnected 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.

H_∞ optimal control of nested systems has only recently been developed. In [1], the author considered the decentralized control of continuous-time time-invariant systems with nested interconnection structure. The discrete-time time-varying version of the optimal control problem was considered in [2].

The theoretical background in this thesis mostly stems off of the results for nested switched systems presented in [3]. The controller synthesis developed in the paper considers the case of a controller with finite memory of past modes. We will then present an extension for a controller with receding horizon modal information, as it was shown for the centralized case in [4].

Lastly, we are interested in seeing how the switched controller performs in a real-world system; more specifically, a system comprised of the HoTDeC hovercrafts developed at the University of Illinois. We would like to see how the system behaves in the scenario of a leader vehicle with followers trailing behind both in simulation and experimentation.

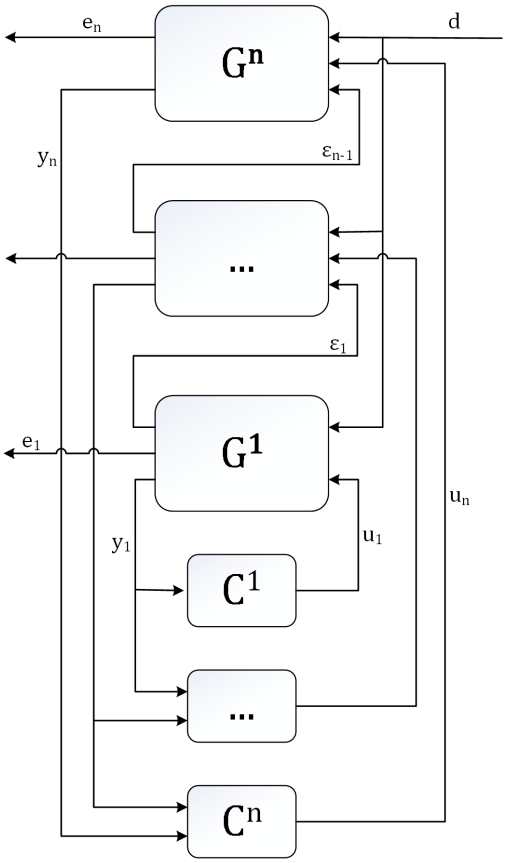


Figure 1.1: Example of a nested interconnection of n-systems.

1.2 Overview

In Chapter 2, we introduce the decentralized switched problem and exact conditions for existence of a controller with finite memory. We then present an extension for a controller with receding horizon modal information.

In Chapter 3, we describe the testbed setup in which the controller was tested. It presents the robots used, along with a brief description of its dynamical model.

Chapter 4 focuses on the controller design process. We also present a simulation model developed in Simulink and some results of the controller developed.

In Chapter 5 we implement the controller into the testbed setup described in Chapter 3. We present some experimental results and compare them with the simulations.

Chapter 6 concludes the thesis and suggests future work.

Chapter 2

Switching Systems

This chapter introduces the switched control problem and conditions for existence of a controller. Background information and the necessary derivations for the analysis of switched systems are provided. Then, an extension of the controller to a discrete-time linear nested system with finite memory and look-ahead horizon is presented. In the end of the Chapter, we discuss an example comparing the performance of controllers with different memory and horizon lengths.

2.1 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, we use $W_\perp$ to denote full column rank matrices satisfying $\mathrm{Im}(W_\perp) = \ker(W)$, $W_\perp^\top W_\perp = I$.

We denote ℓ^n to be the space of infinite indexed sequence of elements as $x = (x(0), x(1), x(2), \ldots)$ 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) which is equipped with the inner-product $\langle x, y\rangle \coloneqq \sum_{t=0}^\infty x(t)^\top y(t)$ and with norm $\sum_{t=0}^\infty |x(t)|_2^2 < \infty$. We denote its norm by $\|\cdot\|$.

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.

2.2 Preliminaries

2.2.1 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 ([5]). The system dynamics are given by

$$\begin{aligned} x(t+1) &= A_{\theta(t)}x(t) + B_{\theta(t)}w(t) \\ z(t) &= C_{\theta(t)}x(t) + D_{\theta(t)}w(t) \end{aligned} \tag{2.1}$$

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 sequences generated by such an automaton are referred to as *admissible* sequences. We denote the set of admissible sequences of length $r \in \mathbb{N}_0$ as $\mathcal{A}_r$.

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

plant

$$\begin{aligned} x(t+1) &= A_{\theta(t)}x(t) + B^w_{\theta(t)}w(t) + B^u_{\theta(t)}u(t) \\ z(t) &= C^z_{\theta(t)}x(t) + D^{zw}_{\theta(t)}w(t) + D^{zu}_{\theta(t)}u(t) \\ y(t) &= C^y_{\theta(t)}x(t) + D^{yw}_{\theta(t)}w(t) \end{aligned} \tag{2.2}$$

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

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

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 $\bar{n} = (n_1, \ldots, n_M)$ and similarly define $\bar{n}^u$ and $\bar{n}^y$.

Since we are interested in nested systems, as in Figure 1.1, 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 $\phi \in \mathcal{N}$.

2.2.2 Path Dependent Systems

Now consider the switched system

$$\begin{aligned} x(t+1) &= A_{\Omega(t)}x(t) + B_{\Omega(t)}w(t) \\ z(t) &= C_{\Omega(t)}x(t) + D_{\Omega(t)}w(t) \end{aligned} \tag{2.3}$$

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

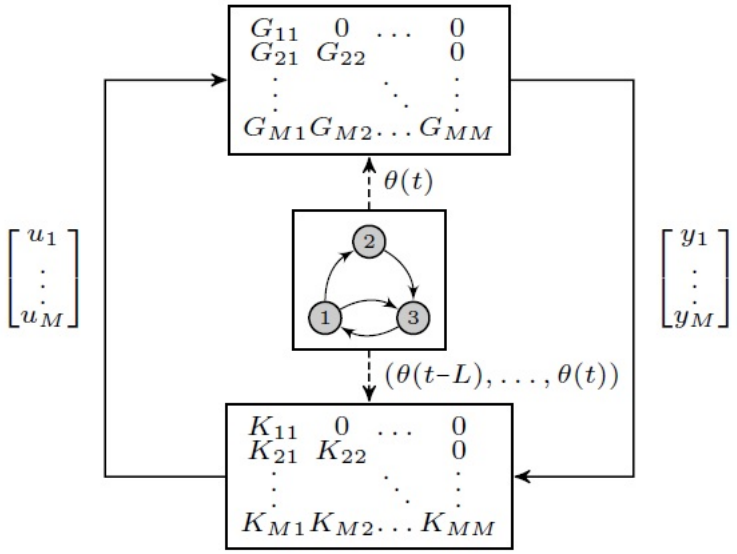


Figure 2.1: Interconnection diagram of a controller with plant. Figure taken from [3].

of systems as *finite-path dependent* systems 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 $\tilde{\mathcal{N}} = \mathcal{A}_{L+1}$. Admissible sequences of length r in the induced automata are denoted by $\tilde{\mathcal{A}}_r^L$. It is not difficult to verify that elements in $\tilde{\mathcal{A}}_r^L$ are equivalent to those of $\mathcal{A}_{r+L}$ for $r > 0$.

For a sequence $\Phi = (\alpha_0, \ldots, \alpha_r) \in \tilde{\mathcal{A}}_{r+1}^L$, there exists an equivalent sequence $(\beta_0, \ldots, \beta_{r+L}) \in \mathcal{A}_{r+L+1}$. For $r > 0$ we define $\bar{\Phi}, \underline{\Phi} \in \mathcal{A}_{r+L}$, $\Phi_\dagger \in \tilde{\mathcal{N}} = \mathcal{A}_{L+1}$, and $\Phi_\star \in \mathcal{N}$ as

$$\bar{\Phi} \coloneqq (\beta_1, \ldots, \beta_{r+L}) \simeq (\alpha_1, \ldots, \alpha_r), \qquad \underline{\Phi} \coloneqq (\beta_0, \ldots, \beta_{r+L-1}) \simeq (\alpha_0, \ldots, \alpha_{r-1}),$$
$$\Phi_\dagger \coloneqq (\beta_0, \ldots, \beta_{r+L}) \simeq \alpha_r, \qquad \Phi_\star \coloneqq \beta_{r+L}.$$

When $r = 0$ these definitions reduce to

$$\bar{\Phi} \coloneqq (\beta_1, \ldots, \beta_L), \qquad \underline{\Phi} \coloneqq (\beta_0, \ldots, \beta_{L-1}),$$
$$\Phi_\dagger \coloneqq (\beta_0, \ldots, \beta_L), \qquad \Phi_\star \coloneqq \beta_L.$$

2.2.3 System Analysis

We wish to have an *exponentially stable* system with ℓ_2 induced norm performance similar to the LTV version for the Kalman-Yakubovich-Popov (KYP) lemma (see [6]). Since switched systems are special cases of LTV systems, we have the following lemma for switched systems.

Lemma 1. The mode-dependent system (2.1) 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-deinite matrices $\{X_\Psi\}_{\Psi \in A_r}$ satisfying

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

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

The following lemma is an extension of the above lemma to path-dependent systems.

Lemma 2. The finite-path dependent system (2.3) with a memory $L \in \mathbb{N}_0$ 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 $\{X_\Psi\}_{\Psi \in \mathcal{A}_{r+L}}$ satisfying

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

for all $\Phi \in \tilde{\mathcal{A}}^L_{r+1}$.

Note that finite-path dependent systems with memory $L_1 \in \mathbb{N}_0$ are also contained in the set of finite-path dependent systems with memory $L_2 > L_1$. Also, suppose the system in (2.3) with memory L_1 has positive-definite scaling matrices $\{X_\psi\}_{\Psi \in \mathcal{A}_{r_1+L_1}}$ satisfying (2) for some $r_1 > 0$. Then we can alternatively choose a memory $L_2 = L_1 + r'$ and $r_2 + r_1 + r'$ for some non-negative integer $r' \leq r_1$ and use the same scaling matrices $\{X_\psi\}_{\Psi \in \mathcal{A}_{r_2+L_2}}$ to describe the same set of inequalities, hence the same stability and performance properties.

2.2.4 Switching control

For the plant (2.2), 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

$$\begin{aligned} x^K(t+1) &= A^K_{\Omega(t)} x^K(t) + B^K_{\Omega(t)} y(t) \\ u(t) &= C^K_{\Omega(t)} x^K(t) + D^K_{\Omega(t)} y(t) \end{aligned} \tag{2.4}$$

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

$$\begin{aligned} A^K_\Psi &\in \mathcal{S}(\bar{n}^K, \bar{n}^K), \qquad & B^K_\Psi &\in \mathcal{S}(\bar{n}^K, \bar{n}^y) \\ C^K_\Psi &\in \mathcal{S}(\bar{n}^u, \bar{n}^K), \qquad & D^K_\Psi &\in \mathcal{S}(\bar{n}^u, \bar{n}^y) \end{aligned} \tag{2.5}$$

where $\bar{n}^K = (n^K_1, \ldots, n^K_M)$. Thus, the resulting controller has a y to u mapping with a lower triangular sparsity structure as depicted in Figure (2.1).

Using a path dependent controller of memory L, with the plant (2.2), the closed-loop system has the following dynamics:

$$\begin{aligned} x^C(t+1) &= A^C_{\Omega(t)} x^C(t) + B^C_{\Omega(t)} w(t) \\ z(t) &= C^C_{\Omega(t)} x^C(t) + D^C_{\Omega(t)} w(t) \end{aligned} \tag{2.6}$$

with $x^C(t) = \begin{bmatrix} x(t) \\ x^K(t) \end{bmatrix}$. At any time t, the closed-loop matrices with depend on the same switching sequence $\Omega(t) \in \mathcal{A}_{L+1}$ as the controller (2.4). Hence, for all sequences $\Psi \in \mathcal{A}_{L+1}$ we can write the closed-loop system matrices as an affine combination of the controller matrices as

$$Q^C_\Psi \coloneqq \begin{bmatrix} A^C_\Psi & B^C_\Psi \\ C^C_\Psi & D^C_\Psi \end{bmatrix} = \left[\begin{array}{cc:c} A_{\Psi_\star} + B^u_{\Psi_\star} D^K_\Psi C^y_{\Psi_\star} & B^u_{\Psi_\star} C^K_\Psi & B^w_{\Psi_\star} + B^u_{\Psi_\star} D^K_\Psi D^{yw}_{\Psi_\star} \\ B^K_\Psi C^y_{\Psi_\star} & A^K_\Psi & B^K_\Psi D^{yw} \\ \hdashline C^z_{\Psi_\star} + D^{zu}_{\Psi_\star} D^K_\Psi C^y_{\Psi_\star} & D^{zu}_{\Psi_\star} C^K_\Psi & D^{zw}_{\Psi_\star} + D^{zu}_{\Psi_\star} D^K_\Psi D^{yw}_{\Psi_\star} \end{array}\right]$$

Then the above can be written as

$$Q_\Psi^C = R_{\Psi_\star} + (U_{\Psi_\star}^C)^\top Q_\Psi^K V_{\Psi_\star}^C \tag{2.7}$$

with $Q_\Psi^K = \begin{bmatrix} A_\Psi^K & B_\Psi^K \\ C_\Psi^K & D_\Psi^K \end{bmatrix}$ representing the unknown controller matrices, and the following defined for $\phi \in \mathcal{N}$

$$R_{\Psi_\star} = \left[\begin{array}{cc:c} A_\phi & 0 & B_\phi^w \\ 0 & 0 & 0 \\ \hdashline C_\phi^z & 0 & D_\phi^{zw} \end{array}\right], \quad (U_\phi^C)^\top = \left[\begin{array}{cc} 0 & B_\phi^u \\ I & 0 \\ \hdashline 0 & D_\phi^{zu} \end{array}\right], \quad V_\phi^C = \left[\begin{array}{cc:c} 0 & I & 0 \\ C_\phi^y & 0 & D_\phi^{yw} \end{array}\right]$$

Since the matrix Q_Ψ^K is structured, we can write it as a linear combination of unstructured matrices as follows,

$$Q_\Psi^K = \sum_{i=1}^{M} \begin{bmatrix} \bar{E}_{i-1}^K & 0 \\ 0 & \bar{E}_{i-1}^u \end{bmatrix} Q_{i,\Psi}^K \begin{bmatrix} E_i^K & 0 \\ 0 & E_i^y \end{bmatrix}^\top \tag{2.8}$$

where $Q_{i,\Psi}^K$ for each $i \in \mathcal{J}$ is an unstructured matrix of dimension $\big((n_i^K + n_i^u) + \ldots + (n_M^K + n_M^u)\big) \times \big((n_1^K + n_1^y) + \ldots + (n_i^K + n_i^y)\big)$, and matrices $E_i^\bullet$ and $\bar{E}_i^\bullet$ are defined below for $i \in \bar{\mathcal{J}}$,

$$E_i^\bullet = \begin{bmatrix} I_{n_1^\bullet + \ldots + n_i^\bullet} \\ 0 \end{bmatrix}, \qquad \bar{E}_i^\bullet = \begin{bmatrix} 0 \\ I_{n_{i+1}^\bullet + \ldots + n_M^\bullet} \end{bmatrix}$$

with $\bullet$ denoting one of K, u, or y. Note that the row diumension of above matrices is $n^\bullet$, and that they satisfy $(\bar{E}_i^{\bullet\top})_\perp = E_i^\bullet$ and $(E_i^{\bullet\top})_\perp = \bar{E}_i^\bullet$. We can then write (2.7) as

$$Q_\Psi^C = R_{\Psi_\star} = \sum_{i=1}^{M} (U_{i,\Psi_\star}^C)^\top Q_{i,\Psi}^K V_{i,\Psi_\star}^C \tag{2.9}$$

with

$$U_{i,\phi}^{C} = \left[\begin{array}{cc:c} 0 & (\bar{E}_{i-1}^{K})^{\top} & 0 \\ (\bar{E}_{i-1}^{u})^{\top}(B_{\phi}^{u})^{\top} & 0 & (\bar{E}_{i-1}^{u})^{\top}(D_{\phi}^{zu})^{\top} \end{array}\right]$$
$$V_{i,\phi}^{C} = \left[\begin{array}{cc:c} 0 & (E_{i}^{K})^{\top} & 0 \\ (E_{i}^{y})^{\top}C_{\phi}^{y} & 0 & (E_{i}^{y})^{\top}D_{\phi}^{yw} \end{array}\right]$$

We can define further the following matrices

$$N_{i,\phi}^{y} = \begin{bmatrix} N_{i,\phi}^{y,x} \\ N_{i,\phi}^{y,w} \end{bmatrix} = \begin{bmatrix} (E_{i}^{y})^{\top}C_{\phi}^{y} & (E_{i}^{y})^{\top}D_{\phi}^{yw} \end{bmatrix}_{\perp}$$
$$N_{i,\phi}^{u} = \begin{bmatrix} N_{i,\phi}^{u,x} \\ N_{i,\phi}^{u,z} \end{bmatrix} = \begin{bmatrix} (\bar{E}_{i}^{u})^{\top}(B_{\phi}^{u})^{\top}(\bar{E}_{i}^{u})^{\top}(D_{\phi}^{zu})^{\top} \end{bmatrix}_{\perp}$$

With respect to Lemma 2, the closed loop scaling matrices are denoted by $X_{\Psi}^{C} \in \mathbb{S}_{+}^{n+n^{K}}$, defined for each $\Psi \in \mathcal{A}_{r+L}$ and some appropriately chosen $r \in \mathbb{N}_{0}$. These matrices are partitioned into plant and controller sections as

$$X_{\Psi}^{C} = \begin{bmatrix} X_{\Psi} & X_{\Psi}^{GK} \\ (X_{\Psi}^{GK})^{\top} & X_{\Psi}^{K} \end{bmatrix}, \qquad (X_{\Psi}^{C})^{-1} = \begin{bmatrix} Y_{\Psi} & Y_{\Psi}^{GK} \\ (Y_{\Psi}^{GK})^{\top} & Y_{\Psi}^{K} \end{bmatrix} \tag{2.10}$$

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 $i \in \bar{\mathcal{J}}$,

$$\begin{aligned} Z_{i,\Psi} &\coloneqq \left\{ X_{\Psi} - X_{\Psi}^{GK}\bar{E}_{i}^{K}\left((\bar{E}_{i}^{K})^{\top}X_{\Psi}^{K}\bar{E}_{i}^{K}\right)^{-1}(X_{i}^{GK}\bar{E}_{i}^{K})^{\top} \right\}^{-1} \\ &= Y_{\Psi} - Y_{\Psi}^{GK}E_{i}^{K}\left((E_{i}^{K})^{\top}Y_{\Psi}^{K}E_{i}^{K}\right)^{-1}(Y_{\psi}^{GK}E_{i}^{K})^{\top} \end{aligned} \tag{2.11}$$

The following lemma is the discrete time equivalent of the Elimination Lemma presented in [7]

Lemma 3. Given $Z \in \mathbb{S}_{+}^{n}$, $\tilde{Z} \in \mathbb{S}_{+}^{m}$, $R \in \mathbb{R}^{n \times m}$, and matrices $\{U_i\}_{i=1}^{M}$ and $\{V_i\}_{i=1}^{M}$ with column dimensions n and m respectively, satisfying

$$\ker(U_1) \subset \ker(U_2) \subset \cdots \subset \ker(U_M)$$
$$\text{and} \quad \ker(V1) \supset \ker(V_2) \supset \cdots \supset \ker(V_M)$$

the following inequality in the unstructured variables $\{Q_i\}_{i=1}^M$

$$Z - \left(R + \sum_{i=1}^{M} U_i^\top Q_i V_i\right)^\top \tilde{Z} \left(R + \sum_{i=1}^{M} U_i^\top Q_i V_i\right) \succ 0 \tag{2.12}$$

has a solution if and only if the following holds

$$\begin{bmatrix} U_{i+1\perp} & 0 \\ 0 & V_{i\perp} \end{bmatrix}^\top \begin{bmatrix} \tilde{Z}^{-1} & R \\ R^\top & Z \end{bmatrix} \begin{bmatrix} U_{i+1\perp} & 0 \\ 0 & V_{i\perp} \end{bmatrix} \succ 0, \quad \text{for } i \in \bar{\mathcal{J}}. \tag{2.13}$$

Now, the next lemma develops necessary conditions for existence of the controller by using Lemmas 2 and 3

Lemma 4. Consider the system (2.2). There exists a finite path dependent controller (2.4) structured as (2.5) which stabilizes this system and achieves performance $\|w \mapsto z\| < 1$ if and only if there exists an $L \in \mathbb{N}_0$ and positive definite $\{X_\Psi^C\}_{\Psi \in \mathcal{A}_{r+L}}$ such that corresponding $\{Z_{i,\Psi}\}_{i=0}^M$ (defined by (2.10) and (2.11)) satisfy

$$\begin{bmatrix} N_{\Phi_\star}^u & 0 \\ 0 & N_{\Phi_\star}^y \end{bmatrix}^\top \left[\begin{array}{cc:cc} Z_{i,\bar{\Phi}} & 0 & A_{\Phi_\star} & B_{\Phi_\star}^w \\ 0 & I & C_{\Phi_\star}^z & D_{\Phi_\star}^{zw} \\ \hdashline A_{\Phi_\star}^\top & (C_{\Phi_\star}^z)^\top & Z_{i,\bar{\Phi}}^{-1} & 0 \\ (B_{\Phi_\star}^w)^\top & (D_{\Phi_\star}^{zw})^\top & 0 & I \end{array}\right] \begin{bmatrix} N_{\Phi_\star}^u & 0 \\ 0 & N_{\Phi_\star}^y \end{bmatrix} \succ 0 \tag{2.14}$$

for all $i \in \bar{\mathcal{J}}$ and $\Phi \in \mathcal{A}_{L+1}$.

The conditions above are not sufficient because for some real L, the existence of $Z_{i,\Psi}$ does not directly imply the existence of a X_Ψ^C for each $\Psi \in \mathcal{A}_L$.

Lemma 5. For a symmetric matrix $X = \begin{bmatrix} X_1 & X_2 \\ X_2^\top & X_3 \end{bmatrix}$ with invertible $X_1 \in \mathbb{S}^{m1}$, $X_2 \in \mathbb{R}^{m1 \times m2}$, and $X_3 \in \mathbb{S}^{m2}$, we can define the triple $\{Z^a, Z^b, Z^c\}$ with $Z^a \in \mathbb{S}^{m1}$, $Z^b \in \mathbb{R}^{m1 \times m2}$, and $Z^c \in \mathbb{S}^{m2}$, related to X by the following bijective mapping

$$Z^a = X_1^{-1}, \quad Z^b = -X_1^{-1}X_2, \quad Z^c = X_3 - X_2^\top X_1^{-1} X_2$$

The triple then defines the following unique factorization

$$X = \begin{bmatrix} I & 0 \\ -(Z^b)^\top & Z^c \end{bmatrix} \begin{bmatrix} Z^a & Z^b \\ 0 & I \end{bmatrix}^{-1} \tag{2.15}$$

Further $X \succ 0$ if and only if $Z^a \succ 0$ and $Z^c \succ 0$.

$$\begin{aligned} Z^a_{i,\Psi} &\coloneqq (E_i^\top Z_{i,\psi} E_i)^{-1} \\ Z^b_{i,\Psi} &\coloneqq -Z^a_{i,\psi}(E_i^\top Z_{i,\psi} \bar{E}_i) \\ Z^c_{i,\Psi} &\coloneqq \bar{E}_i^\top Z_{i,\psi} \bar{E}_i - (E_i Z_{i,\psi} \bar{E}_i)^\top (E_i^\top Z_{i,\psi} E_i)^{-1} E_i^\top Z_{i,\psi} \bar{E}_i \end{aligned} \tag{2.16}$$

$$Z_{i,\psi} = Z^l_{i,\psi}(Z^u_{i,\psi})^{-1} = (Z^u_{i,\psi})^{-\top}(Z^l_{i,\psi})^\top \tag{2.17}$$

with

$$Z^l_{i,\psi} = \begin{bmatrix} I & 0 \\ -(Z^b_{i,\psi})^\top & Z^c_{i,\psi} \end{bmatrix}, \quad Z^u_{i,\psi} = \begin{bmatrix} Z^a_{i,\psi} & Z^b_{i,\psi} \\ 0 & I \end{bmatrix}. \tag{2.18}$$

Sufficient conditions for the existence of controller synthesis is given by

Lemma 6. Given positive-definite matrices $\{Z_{i,\Psi}\}_{i\in\bar{\mathcal{J}},\Psi\in\mathcal{A}_L}$, we can construct positive-definite $\{X^C_\Psi\}_{\Psi\in\mathcal{A}_L}$ satisfying (2.10) and (2.11) if and only if,

$$\begin{bmatrix} Z^{-1}_{i,\Psi} & I \\ I & Z_{i-1,\Psi} \end{bmatrix} \succeq 0, \qquad \text{rank} \begin{bmatrix} Z^{-1}_{i,\Psi} & I \\ I & Z_{i-1,\Psi} \end{bmatrix} \leq n + n_i^K \tag{2.19}$$

for all $i \in \mathcal{J}$ and $\Psi \in \mathcal{A}_L$. Further the above rank conditions are always satisfied for $n_i^K \geq n$.

Theorem 1. Consider the mode-dependent systems (2.2) along with Assumption 1. There exists a synthesis of a finite-path dependent controller (2.4) which

(i) is structured as (2.5)

(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 $\{Z^a_{i,\Psi}, Z^b_{i,\Psi}, Z^c_{i,\Psi}\}_{i\in\bar{\mathcal{J}},\Psi\in\mathcal{A}_L}$ satisfying the following

$$Z^a_{i,\Psi} \succ 0,\ Z^c_{i,\Psi} \succ 0 \text{ for all } i \in \bar{\mathcal{J}},\ \Psi \in \mathcal{A}_L \tag{2.20a}$$

$$\begin{bmatrix} N^u_{i,\Phi_\star} & 0 \\ 0 & N^y_{i,\Phi_\star} \end{bmatrix}^\top \left[\begin{array}{cc:cc} (Z^u_{i,\bar{\Phi}})^\top Z^l_{i,\bar{\Phi}} & 0 & (Z^u_{i,\bar{\Phi}})^\top A_{\Phi_\star} Z^l_{i,\underline{\Phi}} & (Z^u_{i,\bar{\Phi}})^\top B^w_{\Phi_\star} \\ 0 & I & C^z_{\Phi_\star} Z^l_{i,\underline{\Phi}} & D^{zw}_{\Phi_\star} \\ \hdashline \cdot & \cdot & (Z^u_{i,\underline{\Phi}})^\top Z^l_{i,\underline{\Phi}} & 0 \\ \cdot & \cdot & 0 & I \end{array}\right] \begin{bmatrix} N^u_{i,\Phi_\star} & 0 \\ 0 & N^y_{i,\Phi_\star} \end{bmatrix} \succ 0 \tag{2.20b}$$

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

$$\begin{bmatrix} (Z^u_{i,\Psi})^\top Z^l_{i,\Psi} & (Z^l_{i,\Psi})^\top Z^u_{i-1,\Psi} \\ (Z^u_{i-1,\Psi})^\top Z^l_{i,\Psi} & (Z^u_{i-1,\Psi})^\top Z^l_{i-1,\Psi} \end{bmatrix} \succeq 0 \tag{2.20c}$$

$$\operatorname{rank} \begin{bmatrix} (Z^u_{i,\Psi})^\top Z^l_{i,\Psi} & (Z^l_{i,\Psi})^\top Z^u_{i-1,\Psi} \\ (Z^u_{i-1,\Psi})^\top Z^l_{i,\Psi} & (Z^u_{i-1,\Psi})^\top Z^l_{i-1,\Psi} \end{bmatrix} \leq n + n^K_i \tag{2.20d}$$

Now that we have exact conditions for existence of controller synthesis, we can use the next theorem to construct a finite-path dependent controller with memory L and dimensions $n^K_i = n$ for $i \in \mathcal{J}$.

Theorem 2. Given matrices $\{Z^a_{i,\Psi}, Z^b_{i,\Psi}, Z^c_{i,\Psi}\}_{i\in\mathcal{J},\Psi\in\mathcal{A}_L}$ satisfying (2.20a)-(2.20c), corresponding $\{X^C_\Psi\}_{\Psi\in\mathcal{A}_L}$ obtained using (2.10) and (2.11) can be used to obtain the following LMI

$$\begin{bmatrix} \operatorname{diag}\big((X^C_{\bar{\Phi}})^{-1}, I\big) & \left(R_{\Phi_\star} + \sum_{j=i}^M (U^C_{j,\Phi_\star})^\top \tilde{Q}^K_{j,\Phi} \tilde{V}^C_{j,\Phi_\star}\right)(V^C_{i-1,\Phi_\star})_\perp \\ (V^C_{i-1,\Phi_\star})^\top_\perp \left(R_{\Phi_\star} + \sum_{j=i}^M (U^C_{j,\Phi_\star})^\top \tilde{Q}^K_{j,\Phi} \tilde{V}^C_{j,\Phi_\star}\right)^\top & (V^C_{i-1,\Phi_\star})^\top_\perp \operatorname{diag}(X^C_{\underline{\Phi}}, I)(V^C_{i-1,\Phi_\star})_\perp \end{bmatrix} \succ 0 \tag{2.21}$$

in variable $\tilde{Q}^K_{i,\Phi}$ for each $\Phi \in \mathcal{A}_{L+1}$, and solved in the order $i = M, \ldots, 1$.

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.

2.3 Receding Horizon

Now we want to extend the previous result so that our nested switching system also accounts for future modes in the model dynamics, as presented for the centralized case in [4].

We will now define $\Phi^+, \Phi^- \in \mathcal{A}_{r+L+H}$, $\Phi_\dagger \in \mathcal{A}_{r+L+H}$, and $\Phi_0 \in \mathcal{N}$ as

$$\Phi^+ \coloneqq (\beta_1, \ldots, \beta_{r+L+H}), \qquad \Phi^- \coloneqq (\beta_0, \ldots, \beta_{r+L+H-1}),$$
$$\Phi_\dagger \coloneqq (\beta_0, \ldots, \beta_{r+L+H}), \qquad \Phi_0 \coloneqq \beta_{r+L}.$$

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

Now consider the switched system

$$\begin{aligned} x(t+1) &= A_{\Theta(t)}x(t) + B_{\Theta(t)}w(t) \\ z(t) &= C_{\Theta(t)}x(t) + D_{\Theta(t)}w(t) \end{aligned} \tag{2.22}$$

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 $\Theta(t) = \big(\theta(t-L), \ldots, \theta(t), \ldots, \theta(t+H)\big) \in \mathcal{A}_{L+H+1}$

We can modify Lemma 2 to include a look-ahead horizon, as it was shown in [4].

Lemma 7. The finite-path dependent system (2.22) 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 proceed in the same way as the previous section. Lemmas 7 and 3 will give an only having to update the subscripts of the given equations. Retracing the steps of the previous section with Lemmas 7 and 3, we obtain a set of LMIs that give necessary and sufficient conditions for the existence of the path dependent controller.

Theorem 3. Consider the mode-dependent systems (2.2) along with Assumption 1. There exists a synthesis of a finite-path dependent controller (2.4) which

(i) is structured as (2.5)

(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 $\{Z^a_{i,\Psi}, Z^b_{i,\Psi}, Z^c_{i,\Psi}\}_{i\in\bar{\mathcal{J}},\Psi\in\mathcal{A}_{L+H}}$ satisfying the following

$$Z^a_{i,\Psi} \succ 0,\ Z^c_{i,\Psi} \succ 0 \text{ for all } i \in \bar{\mathcal{J}},\ \Psi \in \mathcal{A}_{L+H} \tag{2.23a}$$

$$\begin{bmatrix} N^u_{i,\Phi_0} & 0 \\ 0 & N^y_{i,\Phi_0} \end{bmatrix}^\top \left[\begin{array}{cc:cc} (Z^u_{i,\Phi^+})^\top Z^l_{i,\Phi^+} & 0 & (Z^u_{i,\Phi^+})^\top A_{\Phi_0} Z^l_{i,\Phi^-} & (Z^u_{i,\Phi^+})^\top B^w_{\Phi_0} \\ 0 & I & C^z_{\Phi_0} Z^l_{i,\Phi^-} & D^{zw}_{\Phi_0} \\ \hdashline \cdot & \cdot & (Z^u_{i,\Phi^-})^\top Z^l_{i,\Phi^-} & 0 \\ \cdot & \cdot & 0 & I \end{array}\right] \begin{bmatrix} N^u_{i,\Phi_0} & 0 \\ 0 & N^y_{i,\Phi_0} \end{bmatrix} \succ 0 \tag{2.23b}$$

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

$$\begin{bmatrix} (Z^u_{i,\Psi})^\top Z^l_{i,\Psi} & (Z^l_{i,\Psi})^\top Z^u_{i-1,\Psi} \\ (Z^u_{i-1,\Psi})^\top Z^l_{i,\Psi} & (Z^u_{i-1,\Psi})^\top Z^l_{i-1,\Psi} \end{bmatrix} \succeq 0 \tag{2.23c}$$

$$\operatorname{rank} \begin{bmatrix} (Z^u_{i,\Psi})^\top Z^l_{i,\Psi} & (Z^l_{i,\Psi})^\top Z^u_{i-1,\Psi} \\ (Z^u_{i-1,\Psi})^\top Z^l_{i,\Psi} & (Z^u_{i-1,\Psi})^\top Z^l_{i-1,\Psi} \end{bmatrix} \leq n + n_i^K \tag{2.23d}$$

It is interesting to note that Theorem 3 reduces to Theorem 1 when $H = 0$.

Remark 2. If one desires to obtain decentralized controller for a linear time invariant (LTI) plant, the results obtained above can still be used by simply setting $\mathcal{N} = \{1\}$. That is, the automaton only generates one admissible sequence. As a result, for any memory length L and look-ahead horizon H, there exists only one sequence in $\mathcal{A}_{L+H+1}$. We can then drop the subscripts in Theorem 3, and the controller synthesis will generate only one Q^K.

2.4 Example

Let us consider a two player example with 3-mode switching. The switching automaton is as in Figure 2.1. The corresponding system matrices are chosen as

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

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

$$B_\phi^w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad D_\phi^{yw} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C_\phi^z = \begin{bmatrix} 0.52 \end{bmatrix}, \quad D_\phi^{zw} = 0.5$$

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.

For zero memory length and no look-ahead horizon, 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.

Table 2.1: 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		

Chapter 3

Hardware

In this chapter we present the testbed setup used in this thesis. The main hardware consists of the HoTDeC (Hovercraft Testbed for Decentralized Control) vehicle, which was developed at the University of Illinois at Urbana-Champaign, and the Vision system. Detailed information about the HoTDeC and ealier versions of the system can be found in the dissertation [8] and the article [9]. The Vision system is a network of cameras set up in the lab to keep track of the position of the robots. Further details of the individual system components are presented herein.

3.1 HoTDeC

3.1.1 Body

As shown in 3.1, there are two different types of HoTDeC bodies: 3D printed and precision machined. The 3D printed body is made out of ABS-M30 and weights $565g$, while the machined body is made out of dense Styrofoam and weights $325g$. Both bodies have a diameter of $355mm$.

Each vehicle has five thrusters: one is used to generate lift, and the other four are used to generate forces in the x- and y-direction, as seen in Figure 3.2. We can also use a combination of them to generate a moment and rotate the hovercraft body. For example, if we turn on motors 3 and 4, it generates a thrust force in the positive U_y direction. Turning on motors

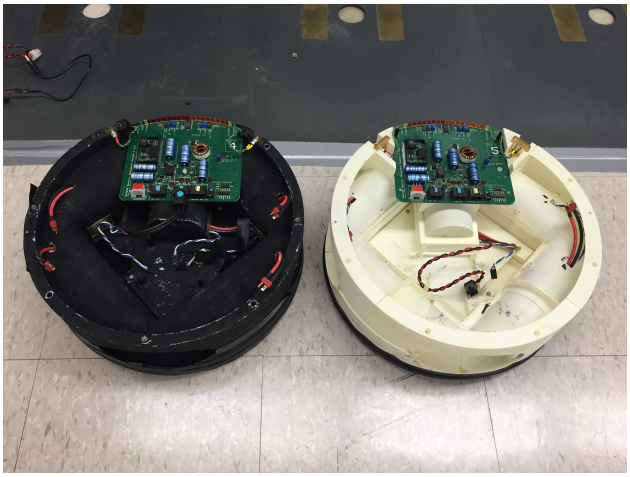


Figure 3.1: Different hovercraft bodies used: Precision machined Styrofoam on the left and 3D printed ABS on the right.

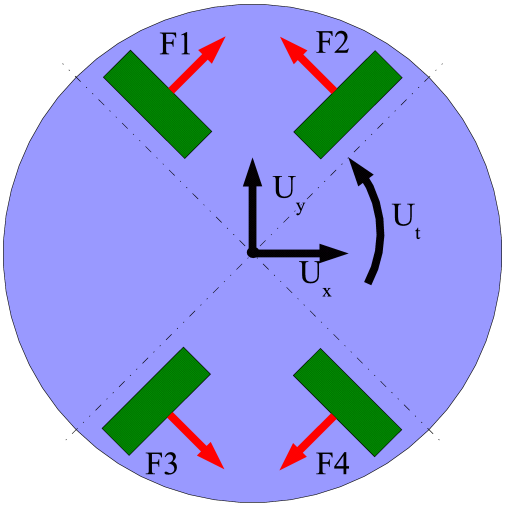


Figure 3.2: Thruster Configuration.

1 and 4 would generate a moment in the positive U_t direction.

3.1.2 Electronics

The HoTDeC consists of four different electronic boards: the Powerboard, the SBC Board, the Isolator Board, and the Motor Board. The Powerboard controls the power distribution to the rest of the hovercraft. The motors are powered by four 3S 2100mAh 25C Lithium Polymer batteries. The SBC board is powered by a single 3S 1320mAh 13 Lithium Polymer battery. The power board handles the step-down to 5V to power the SBC and Isolator

boards, and will shut itself off if the batteries fall below a certain voltage.

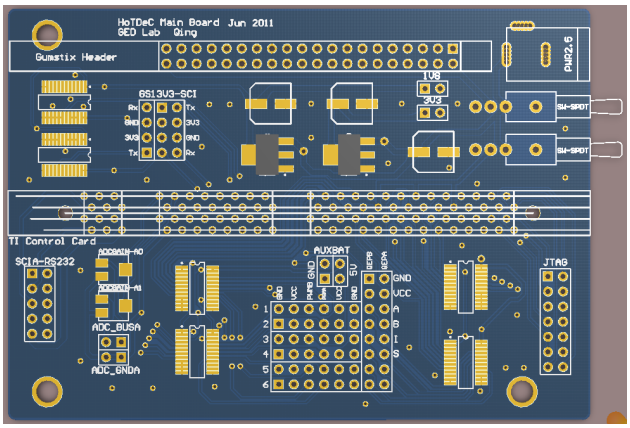


Figure 3.3: SBC Board.

The Single-Board-Computer (SBC) 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, and it supports fully featured real-time embedded Linux operating system called Linaro. For this thesis, the Gumstix is only used to receive commands from the computer and send them to the DSP board via serial connection. However, it could be running its own individual feedback controller loop. The DSP used is a Texas Instruments TMS320F28335, and it's interface contains an input-output pairs for each thruster. It is necessary in the system because the thrusters can reach speeds up to 16,000 RPM. Five of them are currently used to get the Hall effect sensor input and output a PWM signal.

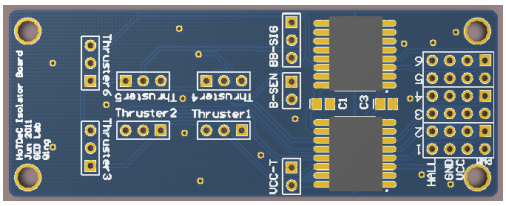


Figure 3.4: Isolator Board.

The Isolator board is designed to isolate the digital circuit from noise. Signals from the DSP are separated from each thruster by this board.

Each motor board includes a hall effect sensor, used to measure the thruster angular velocity, and an H-bridge. The motor blades have two magnets attached, which allows the hall effect sensor to account for each rotation twice.

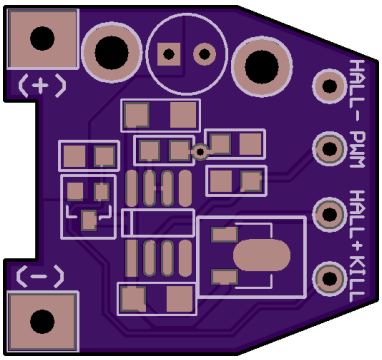


Figure 3.5: Motor Board.

3.2 Dynamical Model

The dynamic model of the HoTDeC is defined by

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Gw(t) \\ y(t) &= Cx(t) + Dv(t) \end{aligned} \tag{3.1}$$

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

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\beta_x}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\beta_y}{m} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{\beta_\theta}{m} \end{bmatrix}, \quad B = G = \begin{bmatrix} 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{1}{m} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{J} \end{bmatrix}, \tag{3.2}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \qquad V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

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 [9].

Table 3.1: Mechanical parameters of the HoTDeC.

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

3.2.1 Motor Dynamics

Each thruster in the hovercraft is controlled via the TI F28335 ControlCARD. The DSP board runs 5 independent proportional integral (PI) loops to control the speed of each motor.

We assume that the relationship between force F and thruster angular velocity ω is $\omega^2 = 2560000 \times F$. The thruster is then modeled as

$$\begin{aligned} \dot{x} &= -8.6x + u \\ w^2 &= 20868760x \end{aligned} \tag{3.3}$$

with

$$F = \begin{cases} 8.1518x - 0.02 & \text{if } x > 0.00246 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Each motor can generate up to $1.2\,N$ of thrust force. The PI loop that controls the voltage supplied to each motor is

$$u = k_P e + k_I \sum_{i=0}^{t} e \cdot \Delta h, \quad k_P = 0.0003, k_I = 0.00015 \tag{3.5}$$

where $e =$ *desired speed*$-$*current speed*, and Δh is the time between samples. A detailed explanation regarding the dynamic model and the controller is presented in [10].

3.3 Network

3.3.1 Vision System

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, as shown by Figure 3.6. A computer processes the information from each of the six cameras, and another computer merges the information to remove any duplicate data from the overlapping areas and broadcasts the data to a network in the lab.

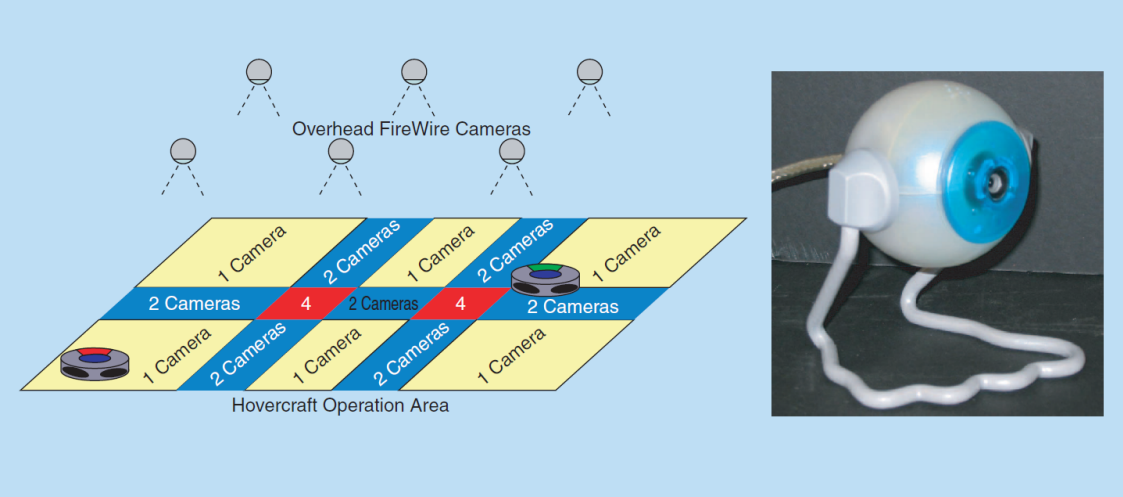


Figure 3.6: Schematic of the Vision system from [9]. On the left is a depiction of the camera coverage and on the right is a picture of one of the cameras used.

The Vision system differentiates between different hovercrafts by using the patterns on their tops, as seen in Figure 3.7. Not only is the pattern used to detect which hovercraft is present, but also to determine which direction the hovercraft is pointing.

3.3.2 Communication System

We use a messaging system called ZeroMQ (ZMQ) to communicate with the vehicles. This system uses two different basic message patterns in the communication layer. One is a Publish-Subscribe structure in which the publisher is constantly broadcasting messages to the network and the subscribers can listen to them at any point in time. The other is a Request-Reply structure in which the requester sends a message and the replier has to

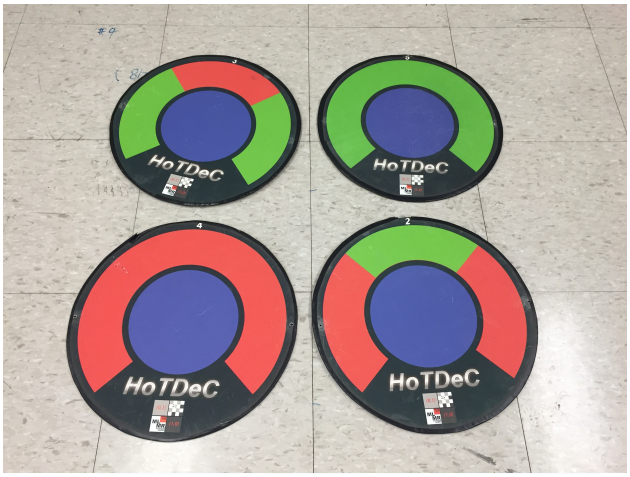
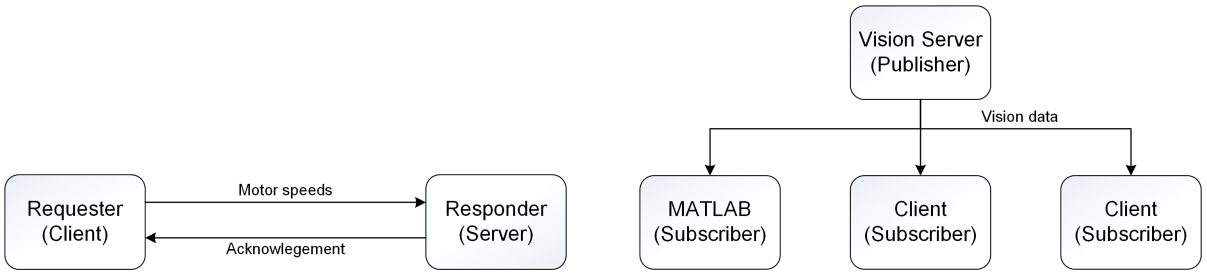


Figure 3.7: Different patterns.

respond back. We use the Publish-Subscribe structure for the Vision system messages and the Request-Reply structure for communicating between the computer and the hovercrafts.



(a) Request-Reply structure. (b) Publisher-Subscriber structure.

Figure 3.8: Two examples of the basic message patterns in ZeroMQ.

Chapter 4

Controller Design

In this chapter the design procedure of the decentralized switched controller controller presented in Chapter 2 is detailed. We also include a MATLAB simulation of the decentralized controller.

4.1 Controller

Our goal is to have the hovercrafts follow each other in a platoon of vehicles, as with the system depicted by Figure 1.1. Each vehicle will try to keep a set distance from the one in front of it, with the first vehicle being told to follow a user defined path. First, to set up our controller, we have to generate the state space system which describes the nested loop, following the form of Equation 2.2.

We implemented a Kalman Filter to reduce the noise from the camera data. With this in mind, the nested plant description assumes that full state information is being provided to the controller. More information on the Kalman Filter will be discussed further on this Chapter.

Using the model dynamics presented in Section 3.2, and the assumption that we have full state feedback, let $\varepsilon_i \in \mathbb{R}^{n_i}$ be the measured position vector of the i-th robot that is available to its followers. Then, if we desire that each hovercraft stays a certain distance $r_i \in \mathbb{R}^{n_i}$ away from its leader, we can choose the performance output $e_i = y_i - \varepsilon_{i-1} - r_i$. The performance output of the first hovercraft will simply be $e_1 = y_1 - r_1$.

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

$$\begin{bmatrix} z_e \\ z_u \\ \hdashline y \end{bmatrix} = \left[\begin{array}{c|c:c} A & B^w & B^u \\ \hline C^z & D^{zw} & D^{zu} \\ \hdashline C^y & D^{yw} & 0 \end{array}\right] \begin{bmatrix} d \\ \hdashline u \end{bmatrix} \tag{4.1}$$

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 hovercraft dynamical model was discretized with a sampling time of $Ts = 0.1$ seconds.

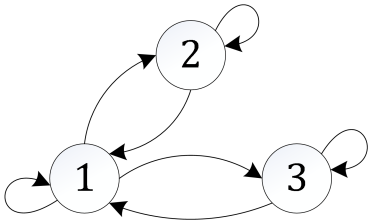


Figure 4.1: Automaton.

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

$$\begin{aligned} z_1 &= \begin{bmatrix} 4x & 2\dot{x} & 4y & 2\dot{y} & 5\theta & 2\dot{\theta} & 0.5u_x & 0.5u_y & 0.6u_\theta \end{bmatrix} \\ z_2 &= \begin{bmatrix} 6x & 2\dot{x} & 3y & 2\dot{y} & 5\theta & 2\dot{\theta} & 0.5u_x & 0.5u_y & 0.6u_\theta \end{bmatrix} \\ z_3 &= \begin{bmatrix} 3x & 2\dot{x} & 6y & 2\dot{y} & 5\theta & 2\dot{\theta} & 0.5u_x & 0.5u_y & 0.6u_\theta \end{bmatrix} \end{aligned} \tag{4.2}$$

where mode 1 represents unobstructed operation, while modes 2 and 3 represented operation near an obstacle in the x- or y-direction respectively. All admissible sequences in our system can be generated by the automaton in Figure 4.1. Note that the weighting vectors above are for a single hovercraft. For the nested system of multiple hovercrafts, these vectors will be

expanded accordingly, with all hovercrafts subjected to the same weights.

CVX, a MATLAB-based toolbox for convex optimization ([11]), was used to solve the LMIs in Theorem 3.

4.1.1 Estimator

To deal with the noisy input from the Vision system, we have added a Kalman Filter using the dynamics from Section 3.2. To generate the filter gain we used the covariance matrices,

$$Q = \mathbb{E}\left[w_n w_n^\top\right] = \begin{bmatrix} \sigma_F^2 & 0 & 0 \\ 0 & \sigma_F^2 & 0 \\ 0 & 0 & \sigma_\tau^2 \end{bmatrix}, \quad R = \mathbb{E}\left[v_n v_n^\top\right] = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \tag{4.3}$$

with

$$\begin{aligned} \sigma_x &= 2.5\, mm, & \sigma_F &= 0.0234\ N \\ \sigma_y &= 2.23\, mm, & \sigma_\tau &= 0.0237\ N \\ \sigma_\theta &= 0.0206\, rad \end{aligned}$$

The Kalman filter runs at 10Hz, the same rate as the controller.

4.2 Simulation

In order to quickly test different controller weightings before deploying the code to the hovercrafts, we have created a Simulink simulation where we modeled the HoTDeC dynamics and the Vision system as described in the previous chapter. Here, a nested structure of four vehicles was considered.

The simulation accounted for not only the dynamics of the hovercrafts but also the dynamics of the thrusters themselves. Figure 4.2 depicts the Simulink schematic for the simulation. Each block named "Hovercraft Simulator #" includes a sub-block that takes in the information from the controller output and simulates the response from the thrusters, as mentioned in Section 3.2.1. Here we also have accounted for motor saturation. This

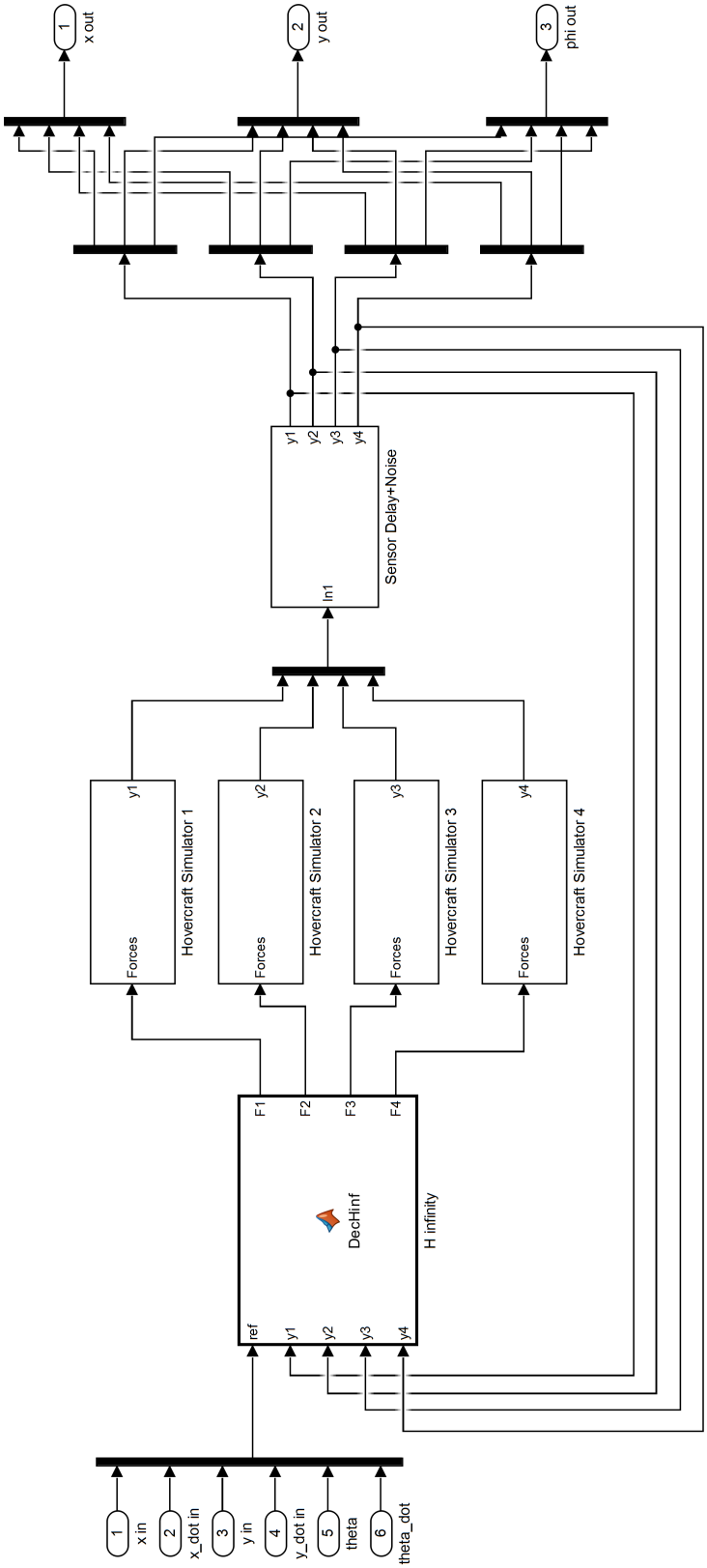


Figure 4.2: Top overview of Simulink model.

sub-block runs the same code that is used on the DSP boards to control the motor speed at the same rate. The information from the motor sub-block is fed into a continuous-time state-space model of the hovercraft dynamics.

The block labeled "Sensor Delay+Noise" simulates the Vision system. It gets new data from the hovercraft box 30 times per second, simulating the refresh rate of the cameras. The Vision system latency is modeled as Gaussian: mean 33.3ms, standard deviation 0.063ms, and the wireless network latency is modeled as Gaussian: mean 2.76ms, standard deviation 0.33ms.

Finally, the "H infinity" block contains the Kalman Filter and the decentralized controller. Both were constructed as described earlier in this chapter. The input to the left of this box is simply the reference signal. The position output from the simulated hovercrafts is saved for later processing.

4.2.1 Results

We tested how well our hovercrafts followed a circular trajectory under different controller memory and horizon lengths. In all test cases the leader was following a $0.6\,m$ wide circular path at a $0.3\,\mathrm{Hz}$ frequency. A LTI case was generated using only the first set of controller weights in Equation 4.2. All controllers were generated at their optimal bounds.

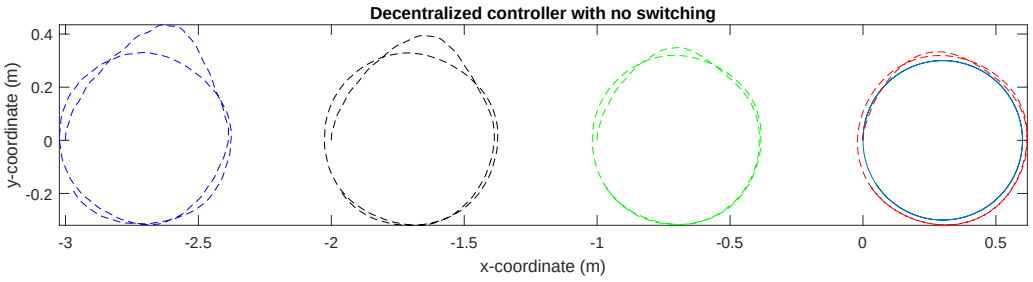


Figure 4.3: Simulated paths of four hovercrafts using the LTI decentralized controller.

In Figure 4.3, 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 1 and look-ahead horizon 1, the trailing robots do a better job at following the one in front (Figures

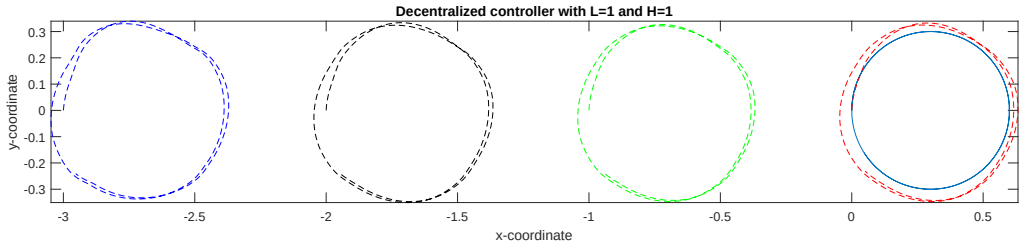
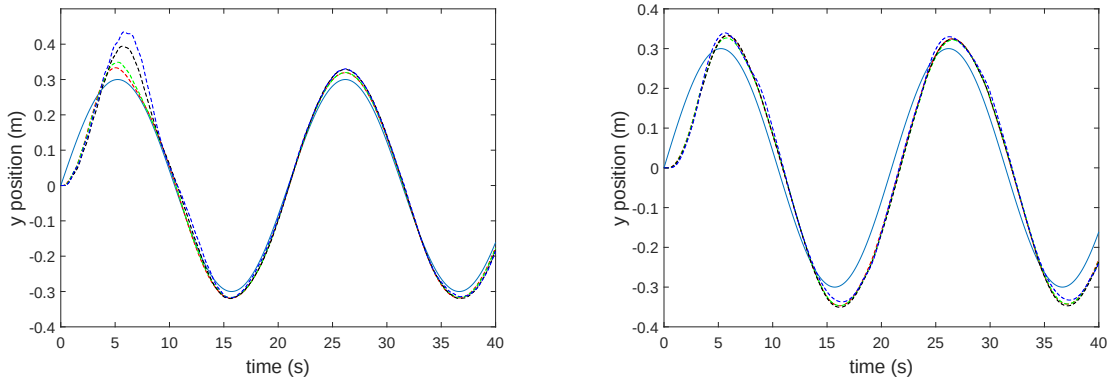


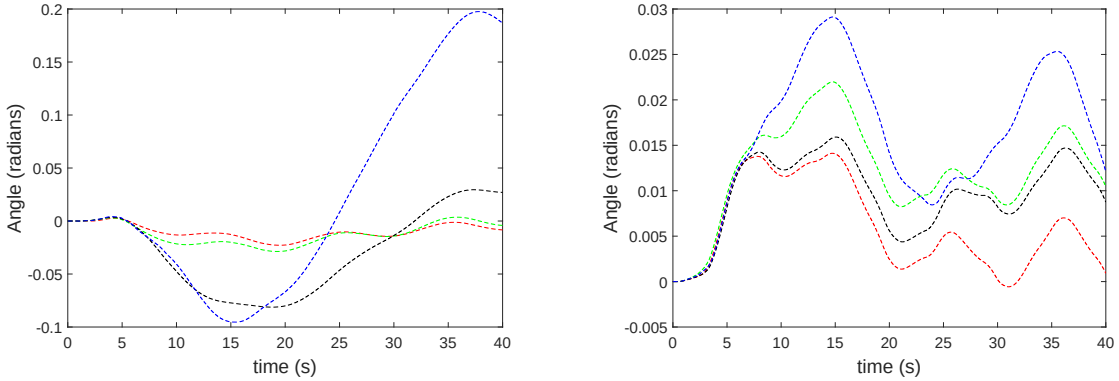
Figure 4.4: Simulated paths of four hovercrafts using the decentralized controller with memory length L=1 and horizon length H=1.

4.4-4.5). As can be seen by Figure 4.6, it is also interesting to note that the hovercrafts do a much better job at keeping their headings facing forward despite the weights for the θ error staying the same.



(a) LTI Controller. (b) L=1, H=1 Controller.

Figure 4.5: Comparison of the simulated y-position.



(a) LTI Controller. (b) L=1, H=1 Controller.

Figure 4.6: Comparison of the simulated angle.

Chapter 5

Hardware Implementation

This chapter presents the testing setup of the hovercraft vehicles and the results from the implementation of an controller with memory length 1 and horizon length 1.

5.1 Setup

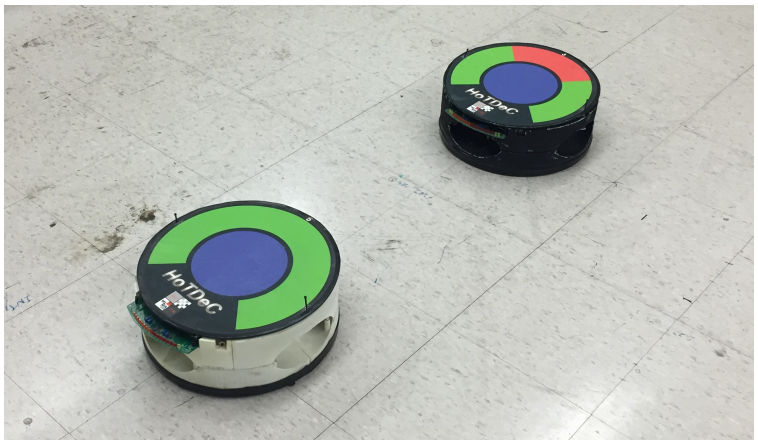


Figure 5.1: Experimental setup of the hovercrafts. The white vehicle is following a circular trajectory while the black vehicle is trying to follow it.

We tested the controller using 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 $0.3\,\mathrm{Hz}$. We have tested a controller with memory $L = 1$ and look-ahead horizon $H = 1$, using the same setup as described in the previous Chapter.

To simplify the process of testing the vehicles and collecting data, the Simulink Desktop Real-Time toolbox was used. As the name suggests, with this toolbox we can run Simulink files in real-time, so the same "H infinity" block used in Section 4.2 can be used to control the hovercrafts in the lab. This simulink model is shown below.

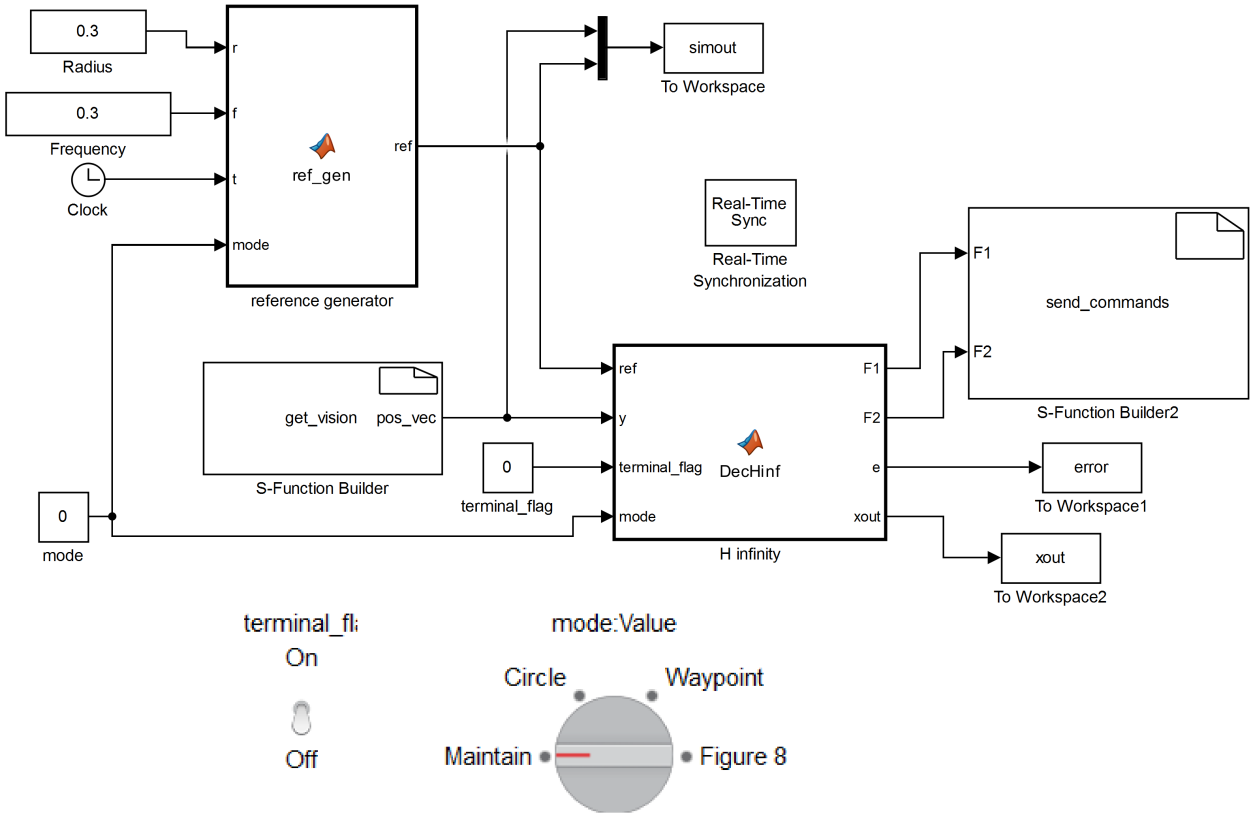


Figure 5.2: Simulink model used in the experiment.

The "get_vision" block receives the hovercrafts' position from the Vision Server, while the "send_commands" block talks with each HoTDeC vehicle. Both blocks were written in C and compiled with MEX so they could be used inside Simulink. The "terminal_flag" switch on the bottom left enable us to start and stop sending commands to the hovercrafts, while the "mode" switch allows us to select between different reference trajectories.

5.2 Results

Main results from the hovercraft experiment are presented in the plots below. It should be noted that in Figures 5.3 and 5.4, 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 figures, the x- and y-positions are the estimated output from the Kalman Filter.

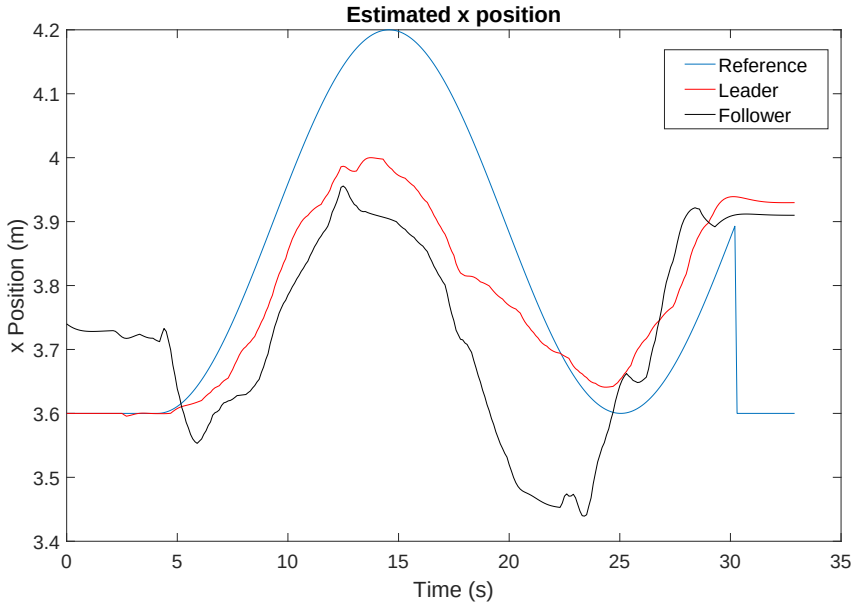


Figure 5.3: Estimated x position of both hovercrafts with a decentralized controller L=1, H=1. The blue line indicates the reference signal.

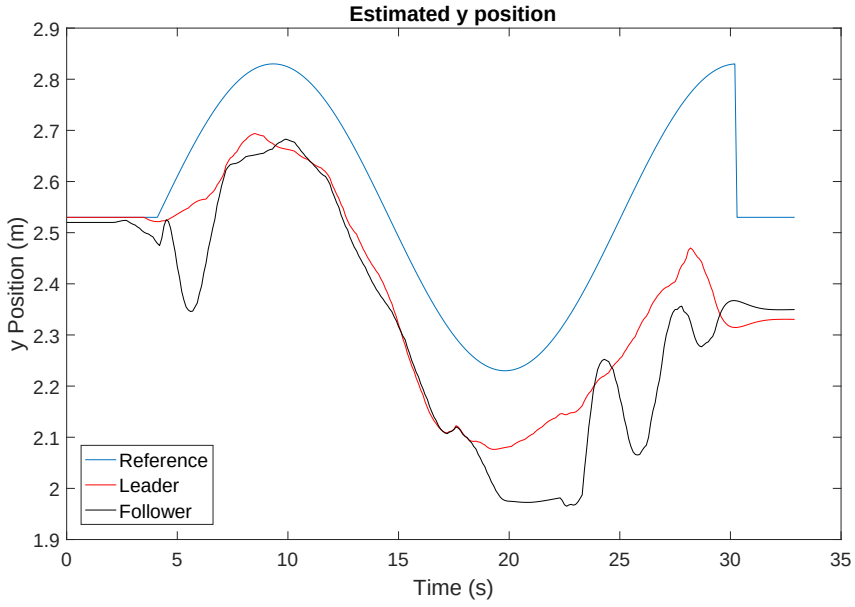


Figure 5.4: Estimated y position of both hovercrafts with a decentralized controller L=1, H=1. The blue line indicates the reference signal.

We can see that while the leader did not follow the reference signal as well as the simula-

tions 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.

Chapter 6

Conclusion

In this thesis, 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 simulation, and then with a real-system experiment. MATLAB and the Simulink package were used to simulate a platoon of four hovercrafts traveling in a user defined path. This platoon consisted of a leader vehicle, which was told which path to follow, and three followers. Each of the followers would try to keep a set distance away from the vehicle in front of it. For this thesis, a circular reference path was chosen. Comparing the general LTI case with that of the switched controller with memory L = 1 and look-ahead horizon H = 1, we can see a significant improvement. With the LTI case, small deviations in movement were seen to worsen with each following vehicle. However, with the L=1, H=1 controller, errors were more consistent across the whole platoon. Additionally, the hovercrafts using the decentralized, look-ahead controller were seen to do a better job of keeping their headings forward, and not twisting as much as with the LTI.

Testing the system experimentally with two HoTDeC hovercrafts resulted in errors less promising than the simulated system. However, this does not mean that the controller was unsuccessful. Just as in the MATLAB simulation, these leader hovercraft was told to travel in a circle with its follower trailing behind. There were many outside factors contributing to the discrepancy between the reference path and the vehicle path, such as noise and spotty

coverage of the Vision system. Still, the follower is seen to approach the motion of its leader with the L=1, H=1 controller.

Overall, the results of this thesis have proven that a discrete-time linear nested system controller with finite memory and look-ahead horizon can be a useful tool. The MATLAB 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 improve the continuity between the experimental and simulation results. For example, the Vision system could be improved to better track the vehicles and certain communication components in the hovercrafts can be updated. For future works, the real-system implementation can be further expanded to include more vehicles.

References

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

[2] 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.

[3] 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.

[4] 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.

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

[6] G.E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*. Texts in Applied Mathematics. Springer New York, 2005.

[7] C. Scherer. A complete algebraic solvability test for the nonstrict lyapunov inequality. *Systems & Control Letters*, 25(5):327 – 335, 1995.

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

[9] 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.

[10] J. Rubel. Design and control of hovercraft over a network. Master's thesis, University of Illinois at Urbana-Champaign, 2004.

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