Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Robust Decentralized Switching Control of UAVs using UWB-based Localization and Cooperation ⋆

Joao P. Jansch-Porto * **Geir E. Dullerud** *

* *Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801 USA (e-mail: {janschp2, dullerud}@illinois.edu).*

Abstract: In this paper, we implement a switched decentralized controller, along with a proposed communication protocol, to control a nested multi-agent system without the need for a centralized processing node. More specifically, we apply a recently developed method for switched systems synthesis, which gives exact conditions for existence of a block-lower triangular path-dependent controller with ℓ_2-induced norm performance. The synthesis conditions are given in the form of a semidefinite program (SDP), which is computed offline for a predefined switching sequence. Each robot is equipped with a ultra-wideband (UWB) unit, which allows it to both estimate its position and communicate with other robots.

Keywords: Control of switched systems, Decentralized control, Flying robots, Multi-vehicle systems, Networked robotic systems, Robust control applications.

1. INTRODUCTION

Our main focus of this paper is the application of decentralized control to robotic nested systems with switching dynamics. Many current multi-vehicle systems require a centralized computer for communication and/or coordination between robots, with formation paths being pregenerated offline or computed on a central node that has access to the global system state. In this paper we present a ultra-wideband (UWB) based system that enables us to both localize robots in 3D space and send messages between vehicles. Along with this, we implement the recently developed robust decentralized switching control synthesis methods (Jansch-Porto and Dullerud, 2017, 2018) to handle the stabilization and coordination of the vehicles.

In recent year, UWB has been gaining traction as mostly a indoor positioning system, with applications both in research (Preiss et al., 2017; Ledergerber et al., 2015) and in industry (Pozyx, 2019; Decawave, 2019). ultra-wideband chips have also been included in the latest iteration of iPhones (Lloyd, 2019), bringing the technology to a much wider audience. However, most of the above applications are only for obtaining some estimate of the position of a robot in the environment, and not for sharing information between robots. In Broecker et al. (2018), the authors use UWB to obtain the relative distance between robots, but it still requires external information and has no guarantees in the controller performance. Hence, the controller presented, along with the UWB hardware, allows us to have multiple unmanned aerial vehicles (UAVs) moving in coordination, without the need for any central computer. Additionally, since the UWB unit can act as both a receiver or transmitter, we can use it as a platform agnostic messaging system.

⋆ The authors were partially supported by NSF Grant 1629949.

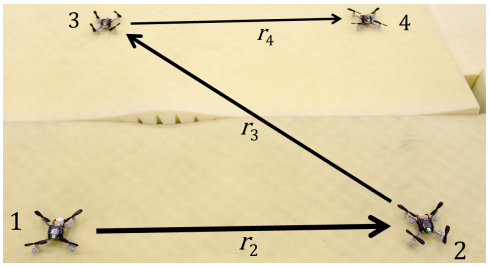

Fig. 1. Example of a leader-follower setup. The bottom-left quadcopter follows a reference signal r_1, while each other robot maintains a distance r_i away from robot $i-1$. The position y_i and reference r_i of each robot are broadcast downstream to other vehicles using UWB.

The synthesis method used in this paper assumes the plant has a nested structure. Nested systems are characterized by interconnected subsystems with unidirectional flow of information, as depicted in Fig. 1, or systems where the actions of one agent affect the dynamics of other agents downstream. Examples of nested systems can include multi-agent setups, such as platooning vehicles (Hedrick et al., 1994), or the interaction between wind turbines in a wind farm (Buccafusca et al., 2019).

The controller generated also allows the plant to vary its parameters based on some switching sequence. Switched systems are widely studied in control theory due to their appearance in many different contexts. In robotics, switched systems are normally used to represent changes in the dynamical model; this includes linearization of nonlinear dynamics around different operating points (Arifianto and Farhood, 2015), network communication modeling (Seiler and Sengupta, 2005), or switching between different behaviors (Egerstedt, 2000). Switched systems can also be used to represent changes in the performance objectives,

which is in a manner similar to discrete parameter gain scheduling (Apkarian et al., 1995).

For the controller synthesis, we will consider the H_∞-type cost criteria, also known as disturbance attenuation or root-mean square gain. The controller presented is restricted to have the same block-lower triangular structure as the plant dynamics. Further, the controller has *memory* of finitely many past modes and access to a finite preview of the switching sequence. To demonstrate what it means to have a finite *preview horizon* we present the following example. Consider a vehicle moving on a highway with a Light Detection and Ranging (LIDAR) unit. We switch between two operating modes: *unobstructed*, when there are no other cars close by, and *obstructed*, for when there other cars in close proximity. If no other vehicles are detected by the LIDAR, we set our future modes to be *unobstructed*. However, if the LIDAR detects other objects, we then set the future modes to be *obstructed*. As such, we can think of our *preview horizon* as how far ahead our sensors can accurately predict what our behavior will be.

Prior work on distributed systems using state-space techniques include D'Andrea and Dullerud (2003); Farhood et al. (2015), which provide sufficient conditions for controller existence and then explicit construction conditionally. However, explicit state-space solutions to H_∞ optimal control of nested systems have is a recent development. In Scherer (2013), the author considers the decentralized control of continuous-time time-invariant systems with nested interconnection structure, presenting exact conditions for controller synthesis. The discrete-time time-varying version of the decentralized H_∞ optimal control problem was considered in Mishra et al. (2014). In Matni (2014), the author presents a solution for strongly connected and partially nested communication constrains under the H_∞ norm. Other authors have discussed the decentralized control of nested systems (Voulgaris, 2000; Lessard and Lall, 2012; Voulgaris et al., 2000; Fardad et al., 2009), however considering different performance criteria.

2. SWITCHING SYSTEMS

For tutorial purposes, we recall the conditions for controller synthesis, along with relevant notation.

2.1 Notation

For any matrix W, $W_\perp$ denotes full column rank matrices satisfying $\mathrm{Im}(W_\perp) = \ker(W)$, $W_\perp^* W_\perp = I$. We let ℓ^n be the space of infinite indexed sequence of elements $x = (x(0), x(1), x(2), \ldots)$ 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), with norm $\|x\|^2 = \sum_{t=0}^{\infty} |x(t)|^2 < \infty$.

We suppress repeated sub-blocks of symmetric matrices, $\begin{bmatrix} X_1 & X_2 \\ X_2^* & X_3 \end{bmatrix}$, as $\begin{bmatrix} X_1 & X_2 \\ \cdot & X_3 \end{bmatrix}$. Similarly, we write inequalities of the form $W^* G W \succ 0$ as $[\bullet]^* G W \succ 0$. We denote the number of subsystems in a nested setup by M, and use the notation $\mathcal{J} = \{1, \ldots, M\}$ and $\bar{\mathcal{J}} = \{0, \ldots, M\}$. The space of block-lower triangular matrices, $\mathcal{S}\big((m_1, \ldots, m_M), (k_1, \ldots, k_M)\big)$, takes the form

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

where $H_{ij} \in \mathbb{R}^{m_i \times k_j}$ and $H_{ij} = 0$ for $i < j$.

2.2 Preliminaries

A switched system is a multi-model system that allows transitions among operation models, where each mode corresponds to a distinct state-space model (Liberzon, 2012). The dynamics are given by the following *mode-dependent* plant:

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

Here $w(t) \in \mathbb{R}^{n^w}$ is the disturbance input, $u(t) \in \mathbb{R}^{n^u}$ is the control input, $z(t) \in \mathbb{R}^{n^z}$ is the performance output, and $y(t) \in \mathbb{R}^{n^y}$ is the measurement output. The system matrices depend on a switching signal $\theta(t)$. We assume that $\theta(t) \in \mathcal{N} = \{1, \ldots, n_s\}$ for all t, and that the switching signal is governed by a finite-state automaton. The set of *admissible* sequences of length $r \in \mathbb{N}_0$ generated by such an automaton is denoted as $\mathcal{A}_r$.

In the decentralized problem, we have the partitions

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

where $x_i(t) \in \mathbb{R}^{n_i}$, $u_i(t) \in \mathbb{R}^{n_i^u}$, and $y_i(t) \in \mathbb{R}^{n_i^y}$. The dimensions satisfy $n = \sum_{i=1}^M n_i$, $n^u = \sum_{i=1}^M n_i^u$, and $n^y = \sum_{i=1}^M n_i^y$. We 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 with unidirectional flow of information, as in Fig. 1, we can make the following assumption about our system matrices:

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 $\phi \in \mathcal{N}$.

Now consider a controller given by

$$\begin{aligned} x^K(t+1) &= A^K_{\Omega(t)} x^K(t) + B^K_{\Omega(t)} y(t) \\ u(t) &= C^K_{\Omega(t)} x^K(t) + D^K_{\Omega(t)} y(t) \end{aligned} \tag{2}$$

whose system matrices at time t depend on a switching path $\Omega(t) = (\theta(t-L), \ldots, \theta(t), \ldots, \theta(t+H)) \in \mathcal{A}_{L+H+1}$. We refer to these types of systems as *finite-path dependent* systems with *memory* length L and *look-ahead horizon* length H. We can modify such systems to be *mode-dependent* by introducing an induced automaton to reflect the path dependence such that $\tilde{\mathcal{N}} = \mathcal{A}_{L+H}$.

The controller state $x^K(t) \in \mathbb{R}^{n^K}$ is partitioned as $\big[(x_1^K(t))^* \ldots (x_M^K(t))^*\big]^*$ with $x_i^K(t) \in \mathbb{R}^{n_i^K}$, thus satisfying $n^K = n_1^K + \ldots + n_M^K$. Since our goal is to design a controller with block-lower triangular sparsity structure, we have

$$\begin{aligned} A^K_\Psi &\in \mathcal{S}(\bar{n}^K, \bar{n}^K), & B^K_\Psi &\in \mathcal{S}(\bar{n}^K, \bar{n}^y), \\ C^K_\Psi &\in \mathcal{S}(\bar{n}^u, \bar{n}^K), & D^K_\Psi &\in \mathcal{S}(\bar{n}^u, \bar{n}^y), \end{aligned} \tag{3}$$

for every $\Psi \in \mathcal{A}_{L+H+1}$.

The closed-loop generated by (1) and (2) has the form

$$\begin{aligned} x(t+1) &= A_{\Omega(t)}x(t) + B_{\Omega(t)}w(t) \\ z(t) &= C_{\Omega(t)}x(t) + D_{\Omega(t)}w(t) \end{aligned} \tag{4}$$

In view of the synthesis theorem, we define

$\bar{\Phi} \coloneqq (\theta(t-L+1), \ldots, \theta(t+H)) \in \mathcal{A}_{L+H}$,
$\underline{\Phi} \coloneqq (\theta(t-L), \ldots, \theta(t+H-1)) \in \mathcal{A}_{L+H}$,
$\Phi_\dagger \coloneqq (\theta(t-L), \ldots, \theta(t+H)) \in \mathcal{A}_{L+H+1}$, $\Phi_0 \coloneqq \theta(t) \in \mathcal{N}$.

2.3 Conditions for Controller Synthesis

For the controller synthesis, we say that the system achieves attenuation level γ if and only if the system $\{(A_i, \gamma^{-1/2}B_i, \gamma^{-1/2}C_i, \gamma^{-1}D_i)\}_{i\in\mathcal{N}}$ is contractive.

Definition 1. Let $\Gamma = \{\gamma_\phi : \phi \in \mathcal{A}_{L+H+1}\}$ be an indexed collection of positive parameters γ_ϕ. Then the system (4) achieves *path-by-path disturbance attenuation levels* Γ if, for every admissible switching sequence, it satisfies

$$\sum_{t=0}^{\infty} |z(t)|^2 \le \sum_{t=0}^{\infty} \gamma_\phi^2 |w(t)|^2 \tag{5}$$

Then, considering the ℓ_2 performance criteria, we have the following conditions for stability and performance.

Lemma 1. (Essick et al. (2014)). The finite-path dependent system (4) with memory $L \in \mathbb{N}_0$ and look-ahead horizon $H \in \mathbb{N}_0$ is exponentially stable and satisfies $\|w \mapsto z\| < \gamma_\phi$ 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

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

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

The following set of Linear Matrix Inequalities (LMIs) give exact conditions for the existence of the controller.

Theorem 2. (Jansch-Porto and Dullerud (2018)). Consider the mode-dependent systems (1) along with Assumption 1. There exists a synthesis of a *finite-path dependent* controller (2) which

(i) is structured as (3)
(ii) has dimensions $\{n_i^K\}_{i=1}^M$
(iii) achieves closed loop performance $\|w \mapsto z\| < \gamma_\phi$ for all $\phi \in \mathcal{A}_{L+H+1}$

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

$$Z^a_{i,\Psi} \succ 0,\ Z^c_{i,\Psi} \succ 0 \text{ for all } i \in \bar{\mathcal{J}},\ \Psi \in \mathcal{A}_{L+H} \tag{6a}$$

$$[\bullet]^* \left[\begin{array}{cc:cc} (Z^u_{i,\bar{\Phi}})^* Z^l_{i,\bar{\Phi}} & 0 & (Z^u_{i,\bar{\Phi}})^* A_{\Phi_0} Z^l_{i,\underline{\Phi}} & (Z^u_{i,\bar{\Phi}})^* B^w_{\Phi_0} \\ 0 & \gamma_{\Phi_\dagger} I & C^z_{\Phi_0} Z^l_{i,\underline{\Phi}} & D^{zw}_{\Phi_0} \\ \hdashline \cdot & \cdot & (Z^u_{i,\underline{\Phi}})^* Z^l_{i,\underline{\Phi}} & 0 \\ \cdot & \cdot & 0 & \gamma_{\Phi_\dagger} I \end{array}\right] N_{i,\Phi_0} \succ 0$$

$$\text{for all } i \in \bar{\mathcal{J}},\ \Phi \in \mathcal{A}_{L+H+1} \tag{6b}$$

$$\begin{bmatrix} (Z^u_{i,\Psi})^* Z^l_{i,\Psi} & (Z^l_{i,\Psi})^* Z^u_{i-1,\Psi} \\ (Z^u_{i-1,\Psi})^* Z^l_{i,\Psi} & (Z^u_{i-1,\Psi})^* Z^l_{i-1,\Psi} \end{bmatrix} \succeq 0 \tag{6c}$$

$$\operatorname{rank} \begin{bmatrix} (Z^u_{i,\Psi})^* Z^l_{i,\Psi} & (Z^l_{i,\Psi})^* Z^u_{i-1,\Psi} \\ (Z^u_{i-1,\Psi})^* Z^l_{i,\Psi} & (Z^u_{i-1,\Psi})^* Z^l_{i-1,\Psi} \end{bmatrix} \le n + n_i^K \tag{6d}$$

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

3. DECENTRALIZED TESTBED

Now we describe the experimental testbed and hardware used to implement the decentralized controller.

3.1 Crazyflie

The Crazyflie 2.0 is a nano quadcopter developed by Bitcraze AB. Due to its open source design and code, coupled with the ability to add extra hardware modules, it is vastly used as a research and development platform (Bitcraze, 2019). Despite its small size, the Crazyflie has an on-board 9-axis Inertial Measurement Unit, a barometer, a radio unit, and an ARM Cortex-M4 processor.

A UWB unit was added to the nanocopter. While the position estimate provided by the UWB sensor is not as accurate as motion capture solutions, it is able to fully run on the embedded microcontroller present in the quadcopter. We will discuss more on the UWB in the next subsection. Each drone also has a unique set of reflective markers so we can track their ground truth position with a motion capture system.

The position data from ultra-wideband is fused with the data from on-board sensors with an Extended Kalman Filter (EKF). The filter is based on the papers Mueller et al. (2015) and Mueller et al. (2016), and was mostly implemented by the designers of the Crazyflie. The estimator code was modified to improve performance and to report all the system states to the controller

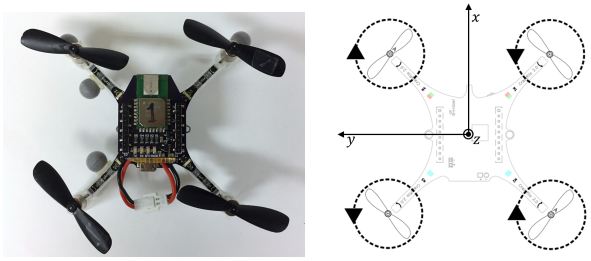

Fig. 2. *Left*: Crazyflie 2.0. The spherical reflective markers used by the Motion Capture system (attached to the frame) and the UWB chip (with the marking "1") can be seen. *Right*: Frame Coordinate System.

3.2 Dynamical Model

The quadcopter dynamics has been widely studied before. For our controller synthesis we will consider the Newton-Euler equations such as the ones presented in Luukkonen (2011). Let $p[m]$ denote the position of the Crazyflie in a Global Cartesian coordinate system, and $\eta[rad]$ be its Euler angles. We have,

$$p = [x\ y\ z]^T, \quad \eta = [\phi\ \theta\ \psi]^T.$$

To simplify the control problem, we assume that the frame is aligned with the global coordinates, and that we directly control the thrust and moments of the quadcopter. Hence,

$$x(t) = \left[p^T\ \dot{p}^T\ \eta^T\ \dot{\eta}^T\right]^T, \quad u(t) = [T\ M_x\ M_y\ M_z]^T$$

where T is the total thrust, and M_x, M_y, M_z are the moments about the x, y, $z-$axis, respectively.

For our problem, we consider the linearized quadcopter dynamics about the stable hovering point $p_e = \dot{p}_e = \eta_e =$

$\dot{\eta}_e = [0\ 0\ 0]^T$, which results on the following continuous-time system matrices:

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & \mathcal{T} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \quad B^u = B^w = \begin{bmatrix} 0_{5\times1} & 0_{5\times3} \\ \frac{1}{m} & 0_{1\times3} \\ 0_{3\times1} & 0_{3\times3} \\ 0_{3\times1} & \mathcal{I} \end{bmatrix}$$

$$\text{where } \mathcal{T} = \begin{bmatrix} 0 & g & 0 \\ -g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{I} = \begin{bmatrix} (\mathbf{I}_{xx})^{-1} & 0 & 0 \\ 0 & (\mathbf{I}_{yy})^{-1} & 0 \\ 0 & 0 & (\mathbf{I}_{zz})^{-1} \end{bmatrix}$$

Here, m is the mass of the robot, g is the acceleration due to gravity, and $\mathbf{I}_{xx}$, $\mathbf{I}_{yy}$, $\mathbf{I}_{zz}$ are the moments of inertia about the x, y, and z axis, respectively. For controller synthesis, we discretized the dynamical model with a sampling time of $Ts = 0.004$ seconds. We also used the full state information from the EKF, so we set $C^y = I_{12\times12}$.

3.3 Indoor Positioning System

In our testbed, aside from the existing Vicon motion capture system, we have developed a localization system using ultra-wideband technology. The UWB units are capable of measuring time-of-flight between other UWB units, enabling us to estimate the position of robots relative to fixed transmitters distributed inside the tested. We have 8 fixed UWB transmitters (also referred to as *Anchors*) transmit messages at a fixed time interval using a Time-division multiple access (TDMA) scheme (see Fig. 4). A receiver in each robot (called *Tag*) calculates the time-difference of arrival (TDOA) between consecutive messages, and estimates its current position using an EKF. Since TDOA algorithms only reliably work inside the convex hull generated by the transmitters, we placed the *Anchors* in the corners of a $10 \times 10 \times 4$m room.

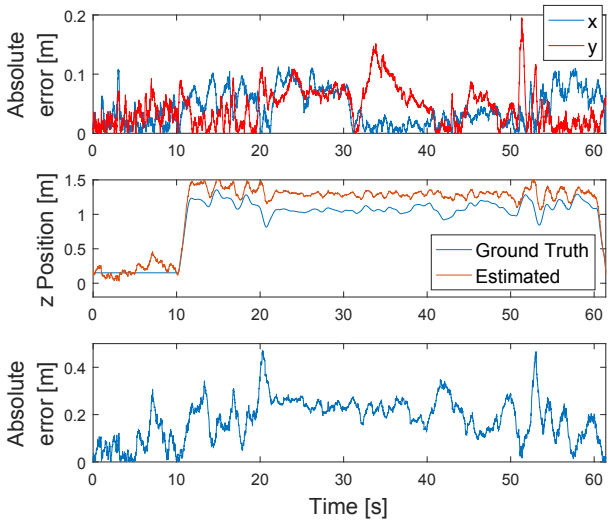

Fig. 3. *Top*: UWB positioning system errors in the x and y axis. *Middle*: Ground truth z-position and Estimate. *Bottom*: Absolute error in the z axis.

We can see a comparison of the performance of UWB as compared to the ground-truth Vicon data in Fig. 3. Since the motion capture system and the UWB setup have slightly different origins, we used the average position before takeoff (at $t = 10$s) to find the offset between both systems. Here we observe that the errors on the x and y axis are, on average, below 5 cm with a few spikes below 18 cm. The errors in the z axis are slightly larger, but on average below 15 cm. This is most likely because before takeoff the quadcopter is sitting slightly below the lowest *Anchors*, and so the initial z estimate is not as reliable as x or y. However we note that, during flight, the z estimate is more stable and has a constant offset to the ground truth data, indicating that if the quadcopter were to start inside the convex hull, the absolute errors would have been smaller. Another possible reason for the higher errors in the z-axis is that, due to the position of our *Anchors*, consecutive messages usually come from two *Anchors* in the height. If we had more variation in z positions, we might have better z estimation.

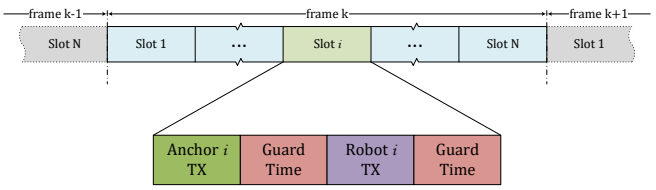

Fig. 4. TDMA frame structure showing the communication protocol.

3.4 Communication Protocol

Since each *Anchor* is transmitting a message at a known interval, we can use the guard time between two consecutive *Anchor* messages to transmit data between robots. In our implementation, robot i broadcasts its current state estimate after it receives a message from *Anchor* i. The resulting messaging scheduling is illustrated in Fig. 4. The protocol could potentially be modified so two robot messages are sent between *Anchor* transmissions. However, that depends on the processor load, *Anchor* message rates, robot message length, among other design choices.

We measured a packet loss rate of 1% when two vehicles are within 2m of each other. However, the reception rate quickly decreases as distance increases as the antenna placement and angle on the Crazyflie 2.0 is not optimal. Packet loss is also affected by other variables, such as proximity to dense solid bodies (such as concrete ground and walls), and transmitter and receiver antennas.

4. EXPERIMENT AND RESULTS

To verify our claims, we implemented the controller and proposed communication protocol on a group of quadcopters. The controller design and results are presented below. The code used to synthesize the controller matrices along with the quadcopter firmware is available on GitHub [1]. The experiments were carried out in part in the Intelligent Robotics Laboratory, University of Illinois.

4.1 Controller Design

Our goal is to have the quadcopters follow each other in a platoon of vehicles. 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. This setup is shown in Fig. 1.

Let $\varepsilon_i \in \mathbb{R}^{n_i}$ be the measured state vector of the i-th robot that is available to its followers. Then, if we desire

[1] `https://github.com/HoTDeC/Robust-Decentralized-Controller`

that each quadcopter maintain a given distance $r_i \in \mathbb{R}^{n_i}$ away from robot $i-1$, we choose the performance output $e_i = r_i + \varepsilon_{i-1} - y_i$, where that of the first vehicle is $e_1 = r_1 - y_1$. Let $r = \begin{bmatrix} r_1^T & r_2^T & \cdots & r_M^T \end{bmatrix}^T$, and similarly define n, d, u, e, y, z_e, and z_u.

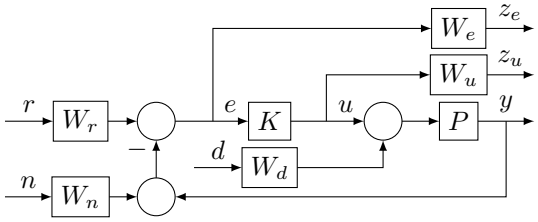

Fig. 5. Block diagram showing interconnection between controller and system, as well as placement of the controller weights.

Figure 5 shows the placement of the weight matrices used to synthesize the feedback controller. Since we are interested in controlling the position of each quadcopter, we set the weighting of the reference input to be $W_r^i = \text{diag}\{1,1,1,1,1,1,0,0,1,0,0,0\}$, $i \in \mathbb{N}$. We set W_n to be the noise variance of each state on the main diagonal. To deal with potential blade differences and mass variations between quadcopters, we added a small static disturbance to the model, with $W_d^i = \text{diag}\{10^{-2},0,0,10^{-4}\}$, $i \in \mathbb{N}$. We also set the controller output weights as $W_u^1 = \text{diag}\{2,40,40,10\}$ for the first vehicle, and $W_u^i = \text{diag}\{2,40,40,5\}$ for robots $i \geq 2$.

For our experiment, the error weighting matrix W_e switches between two different modes so we can have changes in the performance objective. Our plant can switch between the following two modes: in mode 1, we have a small penalty on the x and y coordinates, but larger penalty on the z axis; and in mode 2, all x, y, and z coordinates have similar penalties. In this setup, mode 1 corresponds to takeoff and landing procedures, and mode 2 corresponds to normal flight. Since our Crazyflies are not equipped with external sensors, such as LIDAR or cameras, we implemented a controller with memory $L = 1$, but no look ahead horizon. Therefore, the possible controller switching paths are $\Psi = [11\ 12\ 21\ 22]$. CVX (Grant and Boyd, 2014), was used to solve the controller synthesis LMIs (6).

4.2 Results

The reference input provided to the platoon leader is represented by a dashed line in Fig. 6, while we indicate the jump between modes by the vertical dotted lines. First, for takeoff we set the z velocity to 0.3m/s, stopping at $z = 1$m. Then, we tell the first quadcopter to follow an circular trajectory. Here, the x and y signals are sinusoids with amplitude 0.5m and frequency 0.5Hz. For the landing procedure, we set the z velocity to -0.2m/s.

From the plots in Fig. 6, we can see that the robots successfully follow the leader's reference input. As expected, we see larger tracking errors when the system is in mode 1 because we have smaller x and y penalties during takeoff and landing. However, the robots quickly reduce their tracking errors once we switch to mode 2. While we observe some oscillation, it does not seem to amplify down the chain or cause large instabilities. Better performance and reference tracking could potentially be achieved by having a preview horizon or path-dependent bounds.

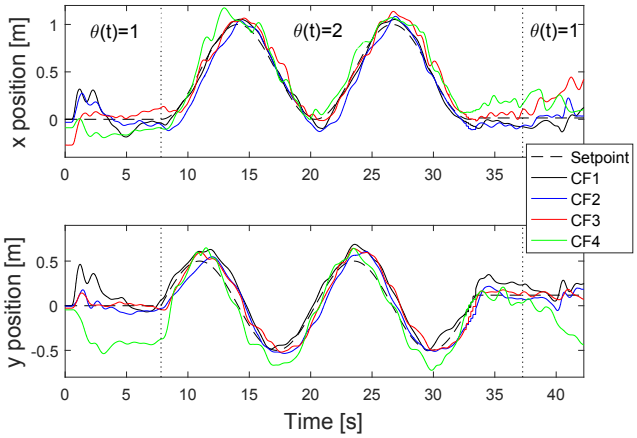

Fig. 6. *Top*: Ground truth x position. *Bottom*: Ground truth y position. The position of quadcopters was shifted by $-r_i, i = 2,\ldots,4$ to simplify visualization.

5. CONCLUSION

In this paper we have implemented a recently developed controller synthesis method, that along with a ultra-wideband unit, allows us to control and coordinate multi-agent systems without a centralized node. The finite-path dependent controller has block lower-triangular structure, provided that the controlled system is also block lower-triangular. Each quadcopter is able to estimate its current position using a UWB unit, which we also use to communicate its state information to other vehicles. Due to the structure of the controller, we are able to control multiple quadcopters with limited communication bandwidth. The proposed system was deployed in a group of quadcopters.

REFERENCES

Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-scheduled h_∞ control of linear parameter-varying systems: a design example. *Automatica*, 31(9), 1251–1261.

Arifianto, O. and Farhood, M. (2015). Optimal control of a small fixed-wing uav about concatenated trajectories. *Control Engineering Practice*, 40, 113–132.

Bitcraze (2019). Used in research. `www.bitcraze.io/research/`. Accessed: 2017-09-10.

Broecker, B., Tuyls, K., and Butterworth, J. (2018). Distance-based multi-robot coordination on pocket drones. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 6389–6394.

Buccafusca, L., Jansch-Porto, J.P., Dullerud, G.E., and Beck, C.L. (2019). An application of nested control synthesis for wind farms. *IFAC-PapersOnLine*, 52(20), 199–204. 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019.

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

Decawave (2019). Partners. `www.decawave.com/partners/`. Accessed: 2019-11-14.

Egerstedt, M. (2000). Behavior based robotics using hybrid automata. In *International Workshop on Hybrid Systems: Computation and Control*, 103–116. Springer.

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

Fardad, M., Lin, F., and Jovanović, M.R. (2009). On the optimal design of structured feedback gains for interconnected systems. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 978–983.

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

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming. `cvxr.com/cvx`.

Hedrick, J.K., Tomizuka, M., and Varaiya, P. (1994). Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6), 21–32.

Jansch-Porto, J.P. and Dullerud, G.E. (2017). Decentralized control with moving-horizon linear switched systems: Synthesis and testbed implementation. In *2017 American Control Conference (ACC)*, 851–856.

Jansch-Porto, J.P. and Dullerud, G.E. (2018). Decentralized control of switched-systems with path-dependent ℓ_2-induced bounds. In *2018 American Control Conference (ACC)*, 502–507.

Ledergerber, A., Hamer, M., and D'Andrea, R. (2015). A robot self-localization system using one-way ultra-wideband communication. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3131–3137.

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

Liberzon, D. (2012). *Switching in Systems and Control*. Systems & Control: Foundations & Applications. Birkhäuser Boston.

Lloyd, C. (2019). Confirmed: Apple developed exclusive tech for the u1 ultra wideband radio. www.ifixit.com/News/inside-the-tech-in-apples-ultra-wideband-u1-chip. Accessed: 2019-11-14.

Luukkonen, T. (2011). Modelling and control of quadcopter. *Independent research project in applied mathematics*.

Matni, N. (2014). Distributed control subject to delays satisfying an $\mathcal{H}_\infty$ norm bound. *arXiv preprint arXiv:1402.1559*.

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

Mueller, M.W., Hamer, M., and D'Andrea, R. (2015). Fusing ultra-wideband range measurements with accelerometers and rate gyroscopes for quadrocopter state estimation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 1730–1736.

Mueller, M.W., Hehn, M., and D'Andrea, R. (2016). Covariance correction step for kalman filtering with an attitude. *Journal of Guidance, Control, and Dynamics*, 1–7.

Pozyx (2019). pozyx. `www.pozyx.io/`. Accessed: 2019-11-14.

Preiss, J.A., Honig, W., Sukhatme, G.S., and Ayanian, N. (2017). Crazyswarm: A large nano-quadcopter swarm. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 3299–3304.

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

Seiler, P. and Sengupta, R. (2005). An h_∞ approach to networked control. *IEEE Transactions on Automatic Control*, 50(3), 356–364.

Voulgaris, P., Bianchini, G., and Bamieh, B. (2000). Optimal decentralized controllers for spatially invariant systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, volume 4, 3763–3768 vol.4.

Voulgaris, P.G. (2000). Control of nested systems. In *2000 American Control Conference (ACC)*, volume 6, 4442–4445.

Appendix A. NOTATION USED IN THE MAIN THEOREM

We denote the space of n-dimensional positive-definite matrices by $\mathbb{S}^n_+$. Due to the desired structured properties of the controller, we define matrices $E^\bullet_i$ and $\bar{E}^\bullet_i$ for $i \in \bar{\mathcal{J}}$

$$E^\bullet_i = \begin{bmatrix} I_{n^\bullet_1+\ldots+n^\bullet_i} \\ 0 \end{bmatrix}, \quad \bar{E}^\bullet_i = \begin{bmatrix} 0 \\ I_{n^\bullet_{i+1}+\ldots+n^\bullet_M} \end{bmatrix}$$

with $\bullet$ denoting one of K, u, or y; the row dimension of above matrices being $n^\bullet$. We have the following kernel space matrices for $i \in \bar{\mathcal{J}}$, $\phi \in \mathcal{N}$,

$$N^y_{i,\phi} = \begin{bmatrix} N^{y,x}_{i,\phi} \\ N^{y,w}_{i,\phi} \end{bmatrix} = \begin{bmatrix} (E^y_i)^* C^y_\phi & (E^y_i)^* D^{yw}_\phi \end{bmatrix}_\perp,$$

$$N^u_{i,\phi} = \begin{bmatrix} N^{u,x}_{i,\phi} \\ N^{u,z}_{i,\phi} \end{bmatrix} = \begin{bmatrix} (\bar{E}^u_i)^* (B^u_\phi)^* & (\bar{E}^u_i)^* (D^{zu}_\phi)^* \end{bmatrix}_\perp.$$

Additionally, we define $N_{i,\phi} := \begin{bmatrix} N^u_{i,\phi} & 0 \\ 0 & N^y_{i,\phi} \end{bmatrix}$. The closed loop scaling matrices are denoted by $X^C_\Psi \in \mathbb{S}^{n+n^K}_+$, defined for each $\Psi \in \mathcal{A}_L$. These matrices are partitioned into plant and controller sections as

$$X^C_\Psi = \begin{bmatrix} X_\Psi & X^{GK}_\Psi \\ (X^{GK}_\Psi)^* & X^K_\Psi \end{bmatrix}, \quad (X^C_\Psi)^{-1} = \begin{bmatrix} Y_\Psi & Y^{GK}_\Psi \\ (Y^{GK}_\Psi)^* & Y^K_\Psi \end{bmatrix}$$

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

$$\begin{aligned} Z_{i,\Psi} &:= \left\{ X_\Psi - X^{GK}_\Psi \bar{E}^K_i \left((\bar{E}^K_i)^* X^K_\Psi \bar{E}^K_i \right)^{-1} (X^{GK}_i \bar{E}^K_i)^* \right\}^{-1} \\ &= Y_\Psi - Y^{GK}_\Psi E^K_i \left((E^K_i)^* Y^K_\Psi E^K_i \right)^{-1} (Y^{GK}_\psi E^K_i)^*, \end{aligned}$$

and the following associated matrices

$$Z^a_{i,\Psi} := (E^*_i Z_{i,\psi} E_i)^{-1}, \; Z^b_{i,\Psi} := -Z^a_{i,\psi} (E^*_i Z_{i,\psi} \bar{E}_i),$$
$$Z^c_{i,\Psi} := \bar{E}^*_i Z_{i,\psi} \bar{E}_i - (E_i Z_{i,\psi} \bar{E}_i)^* (E^*_i Z_{i,\psi} E_i)^{-1} E^*_i Z_{i,\psi} \bar{E}_i.$$

The matrices above give us the factorization $Z_{i,\psi} = Z^l_{i,\psi} (Z^u_{i,\psi})^{-1} = (Z^u_{i,\psi})^{-*} (Z^l_{i,\psi})^*$, where

$$Z^l_{i,\psi} = \begin{bmatrix} I & 0 \\ -(Z^b_{i,\psi})^* & Z^c_{i,\psi} \end{bmatrix}, \text{ and } Z^u_{i,\psi} = \begin{bmatrix} Z^a_{i,\psi} & Z^b_{i,\psi} \\ 0 & I \end{bmatrix}.$$