An Experimental Study of Time Scales and Stability in Networked Multi-Robot Systems

Nathan Michael[1], Mac Schwager[1,2], Vijay Kumar[1], and Daniela Rus[2]

[1] GRASP Laboratory
University of Pennsylvania, 3330 Walnut St
Philadelphia, PA 19106, USA
{nmichael, schwager, kumar}@grasp.upenn.edu

[2] CSAIL
Massachusetts Institute of Technology, 32 Vassar St
Cambridge, MA 02141
rus@csail.mit.edu

Summary. This paper considers the effect of network-induced time delays on the stability of distributed controllers for groups of robots. A linear state space model is proposed for analyzing the coupled interaction of the information flow over the network with the dynamics of the robots. It is shown both experimentally and analytically that control gain, network update rate, and communication and control graph topologies are all critical factors determining the stability of the group of robots. Experiments with a group of flying quadrotor robots demonstrate the effect of different control gains for two different control graph topologies.

1 Motivation, Problem Statement, Related Work

In this work, we investigate the problem of controlling robots over a network in which the time delay for information propagating over the network is not negligible. In experiments with groups of robots it has been observed that the network update rate plays a critical role in the performance and stability of the desired control task [1, 2]. Aggressive control tasks require networks with fast update rates to maintain adequate performance and preserve stability. It is also well known that communication and control graph topologies are important factors in the responsiveness of a network of robots [3]. Networks that are well connected propagate information more quickly, and therefore can support more aggressive control tasks. In this paper we investigate the role of network update time, control gain, and communication and control graph topologies on the stability of a distributed formation controller for a group of robots. We show that the relationship between these factors is more complex than might be expected. For example we show that decreasing the network update time can provoke instability in some cases, and that increasing the connectedness of the control graph can also lead to instability in some cases.

We consider a linear model of robots in a general formation. Each robot is controlled to maintain some prescribed position relative to one or more of the other robots, and the robots communicate over a network defined by a proximity graph. We propagate the robots' state information over the communication network using a well-known networking algorithm called flooding [4, 5]. With this algorithm each robot maintains an outdated estimate of the states of the other robots in the network. We explicitly model the flooding algorithm as a discrete time dynamical system, and we model the robots as continuous time dynamical systems. Using a classical continuous to discrete transformation [6] we derive a model for the robots and the network algorithm in closed loop to investigate stability properties.

We then specialize the model to the case of second order robots using proportional control to maintain a line formation. Analytical results show that stability is not a monotonic function of control gain or network update time, with regimes of instability arising and subsiding as either quantity is increased. Additionally, we observe that for a control graph without loops, the group of robots is stable for all network update times and control gains, but instability can arise in control graphs with loops. In experiments with three quadrotor robots we demonstrate that for a fixed

network update time, increasing control gain can lead to instability, and adding a loop in the control graph can lead to instability.

Our results come from the explicit modeling of both the continuous time dynamics of the robots and the discrete time dynamics of a specific state propagation algorithm. The method we employ to model a continuous time system with discrete network inputs is similar to that used in [7, 8], however the network model in those works does not explicitly consider time delays due to different graph topologies. In [9] the authors use a discrete time model of the coupled robot dynamics and network protocol to analyze a controller for the coordination of ground and aerial vehicles. Also, notions of stability in leader follower formations have been studied without explicitly accounting for the dynamics of the network algorithm, for example in [3]. In the study of consensus, there has been considerable work on quantifying the effects of time delays on the stability of the system, such as in [10, 11] where maximum allowable time delays are derived. In controlling platoons of vehicles in a line, [12] considers the effects of constant time delays between neighbors. There is also a well developed literature on networked control systems with time delays, for which a thorough synopsis can be found in [13].

2 Technical Approach

Let there be n robots of order m in the network, with states $x_i(t) \in \mathbb{X} \subset \mathbb{R}^N$ and trajectories $x_i : \mathbb{R}_{\geq 0} \mapsto \mathbb{X}$ that evolve in continuous time $t \in \mathbb{R}_{\geq 0}$ according to the linear time invariant dynamics

$$\dot{x_i}(t) = A_i x_i(t) + B_i u_i(t).$$

The control input $u_i(t)$ is computed as a function of the states of the other robots in the network. However, robot i only has outdated values of the other robots' states,

$$\hat{x}_i(t) = [\hat{x}_{i1}^{\mathrm{T}}(t) \quad \cdots \quad \hat{x}_{ii}^{\mathrm{T}}(t) \quad \cdots \quad \hat{x}_{in}^{\mathrm{T}}(t)]^{\mathrm{T}},$$

where $\hat{x}_{ij}$ denotes robot i's outdated estimate of the state of robot j. Also, let every robot maintain a vector of the time delay associated with the state estimate of each robot,

$$\tau_i = [\tau_{i1}(t) \quad \cdots \quad \tau_{ii}(t) \quad \cdots \quad \tau_{in}(t)]^{\mathrm{T}},$$

where τ_{ij} is the number of network update steps that have passed since the state value was current, that is $\hat{x}_{ij}(t) = x_j(t - \tau_{ij}(t)T)$, where T is the time it takes for one round of the flooding algorithm to complete. Furthermore, we assume that the robots know their own state without any delay, so that $\hat{x}_{ii}(t) = x_i(t)$ and $\tau_{ii}(t) = 0$.

In the flooding algorithm, each time step represents a cycle of a Time Division Multiple Access (TDMA) networking algorithm in which each robot has a fixed time slot during which to broadcast information. When it is robot j's turn to broadcast, that robot sends a packet containing $\hat{x}_j$ and τ_j. Any robot i within communication range receives the packet and replaces its state estimates with those of robot j that are more current than its own. The time delays are then incremented accordingly. More formally,

$$\hat{x}_{ik}(t) = \begin{cases} \hat{x}_{j^*k}(t-T) & \text{for} \quad i \neq k \\ x_i(t) & \text{otherwise,} \end{cases} \tag{1}$$

and

$$\tau_{ik}(t) = \begin{cases} \tau_{j^*k}(t-T) + 1 & \text{for} \quad i \neq k \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where

$$j^* = \arg \min_{j \in \mathcal{N}_i^{\text{comm}} \cup \{i\}} \tau_{jk}(t-T),$$

and $\mathcal{N}_i^{\text{comm}}$ is the index set of the robots that are neighbors of i in the communication graph.

2.1 Coupled Network and Control System

Consider a controller of the form

$$u_i = \sum_{j \in \mathcal{N}_i^{\text{cont}}} K_{ij}(\hat{x}_{ij}(t) - x_i(t) - d_{ij}), \tag{3}$$

where $\mathcal{N}_i^{\text{cont}}$ is the index set of the robots whose states are required for the computation of robot i's control law. Further divide the index set into subsets $\mathcal{N}_{ik}^{\text{cont}}$ of all control neighbors j that are k hops away from i *in the communication graph*, so that $\mathcal{N}_i^{\text{cont}} = \cup_{k=1}^{n-1} \mathcal{N}_{ik}^{\text{cont}}$ and $\mathcal{N}_{ik}^{\text{cont}} \cap \mathcal{N}_{il}^{\text{cont}} = \emptyset$ for all $k \neq l$. The control input for each robot can be divided into a continuous component $K_{ii}x_i(t)$, where $K_{ii} = \sum_{j \in \mathcal{N}_i^{\text{cont}}} K_{ij}$ and a component that is constant over the time step of the network $\sum_{j \in \mathcal{N}_i^{\text{cont}}} K_{ij}(\hat{x}_{ij}(t) - d_{ij})$. We can rewrite the system as

$$\dot{x}_i(t) = (A_i - B_i K_{ii})x_i(t) + B_i \sum_{k \in \mathcal{N}_i^{\text{cont}}} \sum_{j \in \mathcal{N}_{ik}^{\text{cont}}} K_{ij}(\hat{x}_{ij}(t) - d_{ij}).$$

Now the system looks like a continuous time system with a zero order hold on the input. Without making approximations, we can write a discrete time equation describing the value of the continuous state $x(t)$ at T second intervals as

$$x_i(t+T) = \Phi_i x_i(t) + \Gamma_i B_i \sum_{k \in \mathcal{N}_i^{\text{cont}}} \sum_{j \in \mathcal{N}_{ik}^{\text{cont}}} K_{ij}(\hat{x}_{ij}(t) - d_{ij}).$$

where

$$\Phi_i = e^{(A_i - B_i K_{ii})T} \quad \text{and} \quad \Gamma_i = \int_0^T e^{(A_i - B_i K_{ii})\tau} d\tau,$$

and the exponentials are matrix exponentials [6].

Now define a state vector $x(t) = [x_1(t) \quad \cdots \quad x_n(t)]^{\text{T}}$, which is the concatenation of the states $x_i(t)$, and $X(t) = [x(t) \quad \cdots \quad x(t-(n-1)T)]^{\text{T}}$, which is the concatenation of all the time delayed states up to the maximum number of hops it can take for a message to cross the network, namely $n-1$. This is the state of the coupled network and robots. Because the robot dynamics, controller, and network algorithm are linear, the entire coupled system is linear and can be written

$$X(t+T) = AX(t) + D,$$

where

$$A = \begin{bmatrix} \Phi & C_1 & \cdots & C_{n-1} \\ I_{nm} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I_{nm} & 0 \end{bmatrix},$$

with

$$\Phi = \text{diag}([\Phi_1 \quad \cdots \quad \Phi_n]),$$

$$C_k = [\kappa_{ij}], \quad \kappa_{ij} = \begin{cases} \Gamma_i B_i K_{ij} & \text{if } j \in \mathcal{N}_{ik}^{\text{cont}} \\ 0 & \text{otherwise,} \end{cases}$$

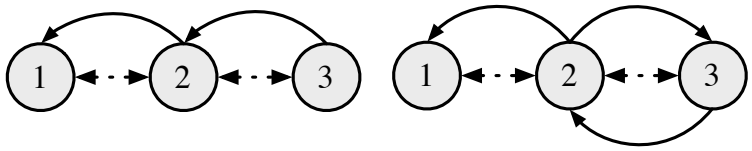
and

$$D = [-(\Gamma_1 B_1 \sum_{j \in \mathcal{N}_1^{\text{cont}}} K_{1j} d_{1j})^T \quad \cdots \quad -(\Gamma_n B_n \sum_{j \in \mathcal{N}_n^{\text{cont}}} K_{nj} d_{nj})^T \quad 0 \quad \cdots \quad 0]^{\text{T}}.$$

The condition for the stability of the closed loop system is that all of the eigenvalues of A lie within the unit circle, that is

$$\|\lambda_i(A)\| \leq 1 \quad \forall i \in \{1, \ldots, n(n-1)\}.$$

The eigenvalues of A are functions of T and K_{ij}, so this gives the condition that we are looking for. One can check this condition straightforwardly by computational means for particular systems and controllers. Unfortunately, the condition is too general to be of use as a rule of thumb, or to give an insight into the trade offs between control gain and network update rate. For this, we seek special cases that are of particular interest.



(a) (b)

Fig. 1. Network and control graphs of the two example systems in Sect. 3.2. Directed solid arrows indicate control feedback. In Fig. 1(a), only K_{12} and K_{23} have non-zero entries while in Fig. 1(b), K_{21}, K_{23}, and K_{32} have non-zero entries. The network graph is the same for both cases and is shown with dashed bi-directional arrows.

3 Results

We briefly detail the theoretical results related to this work and note that a more thorough study is in preparation [14].

3.1 Integrator Robot Model

For the case of robots with integrator dynamics, $\dot{x}_i = u_i$, the system is stable for all positive control gains, $K_{ij} \geq 0$, and for all positive network update times, $T \geq 0$. This can be proved with Geršgorin's discs on A. This surprising fact indicates that an integrator robot model, which is common in the multi-robot control literature, is not rich enough to exhibit what could be considered a fundamental property of networked robotic systems; sensitivity to network time delays.

3.2 Second Order Robot Model

For this reason we turn to the case of a second order robot model ($m = 2$) of the form

$$\ddot{p}_i(t) + b\dot{p}_i(t) + cp_i(t) = u_i,$$

where p_i is the position of the robot in the direction parallel to the line formation, and b and c are the damping and spring constants respectively of the robot (potentially in closed loop with an on-board controller). We use the formation controller with proportional control given by

$$u_i = \sum_{j \in \mathcal{N}_i^{\text{cont}}} k(\hat{p}_{ij}(t) - p_i(t) - d_{ij}),$$

where $\hat{p}_{ij}$ is updated with the flooding algorithm according to (1). Following the presentation in the prior section, $x_i = [p_i \quad \dot{p}_i]^T$ and

$$A_i = \begin{bmatrix} 0 & 1 \\ -c & -b \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad K_{ij} = \begin{bmatrix} k & 0 \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad K_{ii} = \begin{bmatrix} |\mathcal{N}_i^{\text{cont}}|k & 0 \\ 0 & 0 \end{bmatrix}.$$

We now consider example systems to study the effects of the proportional control gain k, network update rate T, and control graph topology.

3 Robots: Tree-Shaped Control Graph

We begin by considering a system with three robots structured with tree-shaped network and control graphs (Fig. 1(a)). As noted above, this system is stable given the control law (3) for any positive control gains and time delays. We use this case as a comparison to the next case where system instabilities are provoked by modifying the control gains and time delays. In Fig. 2(a) the magnitude of the maximum eigenvalue, $\|\lambda_{\max}(A)\|$, is shown for different values of k holding $c = 2.2$, $b = 1$. Note that as expected, the system is stable for all time delays.

3 Robots: Control Graph with Cycle

Figure 2(b) depicts $\|\lambda_{\max}(A)\|$ versus time delay, T, for the control graph shown in Fig. 1(b) in which $\mathcal{N}_1^{\text{cont}} = \emptyset$, $\mathcal{N}_2^{\text{cont}} = \{1, 3\}$, and $\mathcal{N}_3^{\text{cont}} = \{2\}$, and parameter values $c = 2.2$, $b = 1$, and k varied as in the prior case. It is clear that stability for this control graph is dependent on gain and time delays in the system.

4 Experiments

We are interested in examining the phenomena and theoretical results discussed in Sect. 3. We consider aerial robots which we treat as dynamic point model robots. The details of the experimental infrastructure are presented in [15] so here we provide a concise overview.

4.1 Experimental System and Design

The experimental system consists of a Vicon motion capture system [16], three quadrotors sold by Ascending Technologies, GmbH [17] (see Fig. 3), and supporting code integrated via ROS [18]. The environment in which the robots fly is surrounded by a net. The robots require inputs in terms of desired attitudes: roll, pitch, and yaw change (ϕ, θ, $\Delta\psi$,

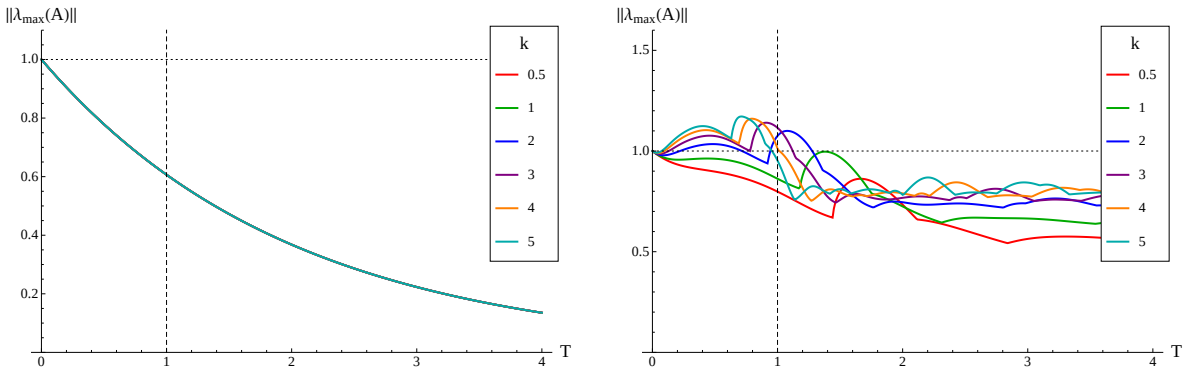


Fig. 2. The magnitude of the maximum eigenvalue, $\|\lambda_{\max}(A)\|$, versus time delays, T, for various control gain values, k, for the systems shown in Fig. 1 with Fig. 2(a) corresponding to Fig. 1(a) and Fig. 2(b) corresponding to Fig. 1(b). The system is stable if $\|\lambda_{\max}(A)\| \leq 1$ (with $\|\lambda_{\max}(A)\| = 1$ shown as a dotted black line). Note that Fig. 2(a) confirms the stability of the system with no cycles for any T and k as $\|\lambda_{\max}(A)\| \leq 1$. Figure 2(b) depicts the behavior observed in prior experimentation that motivates this work where arbitrary selection of k or T may yield an unstable system. Plots are computed by finding the analytic solution to the eigenvalues of A and solving for $\|\lambda_{\max}(A)\|$ from $T = 0$ to $T = 4$ at intervals of 0.01. The delay applied in experiments, $T = 1$, is highlighted by a dashed line.

Fig. 3. The team of three quadrotors used in experiments.

respectively), and thrust, F. We compute and send these values to the processor on the robot at 100 Hz where the inputs are applied at 1 kHz. The values are determined by the following transformations

$$\begin{aligned}
\phi^{\text{des}} &= u_x \sin\psi - u_y \cos\psi + \phi_{\text{trim}} \\
\theta^{\text{des}} &= u_x \cos\psi + u_y \sin\psi + \theta_{\text{trim}} \\
\Delta\psi^{\text{des}} &= k_{p,\,\psi}\left(\psi^{\text{des}} - \psi\right) + \psi_{\text{trim}} \\
F &= F_m + k_{p,\,z}\left(z^{\text{des}} - z\right) - k_{d,\,z}\dot{z} + F_{\text{trim}},
\end{aligned}$$

where trim values serve to compensate for performance variation between robots. The control inputs are defined as

$$u_x = u_i \quad \text{and} \quad u_y = k_{p,\,y}(y^{\text{des}} - y) - k_{d,\,y}\dot{y}.$$

Time delays are introduced into the system by applying a zero-hold on the feedback of the position of the neighboring robots. A hold of $T = 1$ s is applied for all experiments. Communication between robots is performed using TCP communications over 802.11 which introduces latencies on the order of a few milliseconds.

In this work, we present three experiments to highlight the effects of loops in the control graph on the stability of the system. In each experiment, three robots control to maintain a separation distance of 1 m with $b = 1$ and $T = 1$ s. The first robot holds its commanded position (for these experiments $x_1^{des} = 2$ m) while the other robots control based on the network and communication graph. For these preliminary experiments we construct a network that requires only one hop communications. We ensure that the robots do not collide by separating their positions along the y-axis by 1 m (see Table 1). Additional variations in the experiments are summarized in Table 2.

Robot	x	y	z
1	2	-1	2
2	0	0	2
3	-2	1	2

Table 1. Robot initial positions.

Experiment	Graph	k
1	Fig. 1(a)	2
2	Fig. 1(b)	2
3	Fig. 1(b)	1

Table 2. Experiment descriptions and control gains.

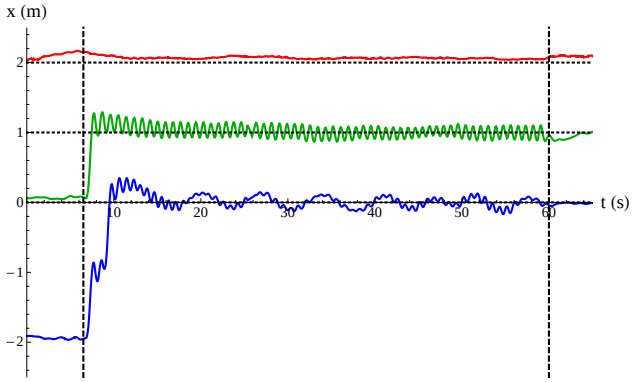
4.2 Experimental Results

Figures 4 and 5 show the results of the three experiments. We tested these cases many times and found similar results over all trials. The most significant outcome from the experiments is the clear instability that occurs in the system with a cycle and $k = 2$. Interestingly, the only difference between Fig. 1(a) and Fig. 1(b) is an additional feedback control term between the second and third robots. Note that this fact is predicted in Fig. 2(b) as $k = 1$ at $T = 1$ yields a stable system while $k = 2$ at the same delay yields an unstable system.

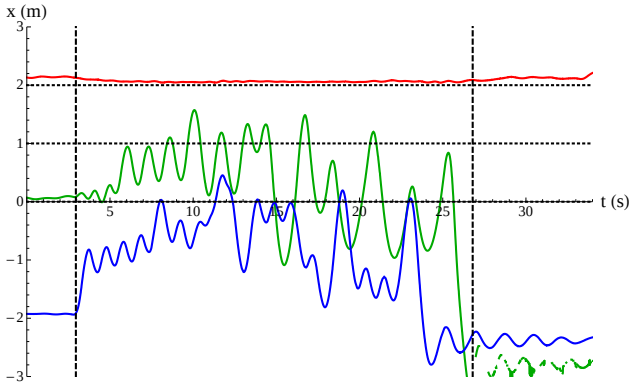
5 Conclusion and Discussion

In this work we consider the role of cycles in the control graph and the introduction of time delays in the communication between robots used for feedback control. We design preliminary experiments based on theoretic analysis that addresses the interplay between network and communication graphs and time delays across the network.

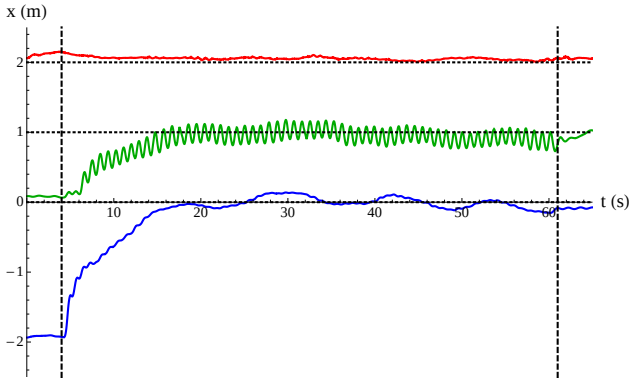
The motivation for this work comes from the observation in prior experiments that the communication rate dictated performance of the algorithm and consequently the ability of the system to control cooperatively. However, increased communication rates may not be feasible due to hardware, bandwidth, and power constraints. A typical response to the problem of lower rates of feedback information is to "slow the system down" by reducing the control gains. However, this results in lower performance and an inability to perform more aggressive maneuvers.



(a) No cycles, $k = 2$

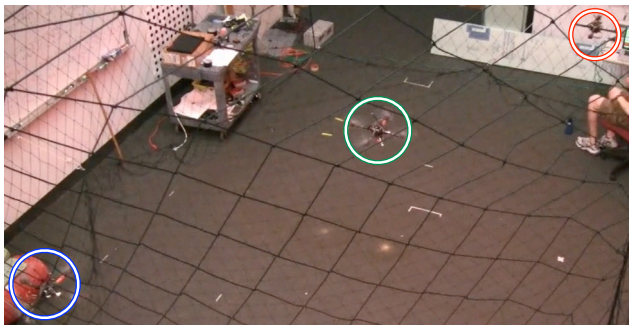


(b) Cycle, $k = 2$

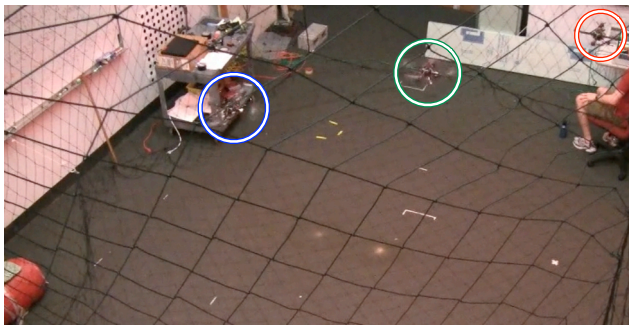


(c) Cycle, $k = 1$

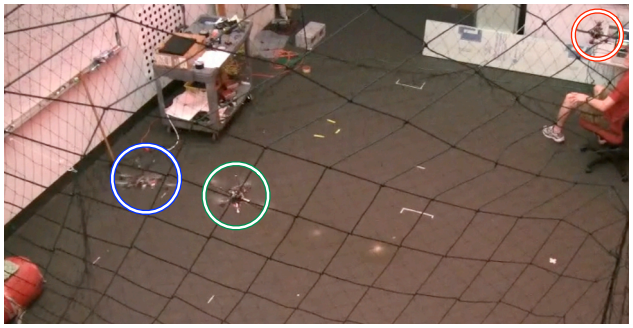
Fig. 4. The x position of each of the robots during the three experiments. Robots 1, 2, and 3 are shown in red, green, and blue respectively. The vertical black dotted lines represent the start and finish of the experiment. Changing the control network while maintaining the control gain results in an unstable system (Fig. 4(b)). Decreasing the gain yields a stable system (Fig. 4(c)) but with considerably different performance to the system without cycles (Fig. 4(a)).



(a)



(b)



(c)

Fig. 5. Images from the experiments. The robots start at an initial configuration (Fig. 5(a)). For stable systems, the robots converge to the desired separation distance (Fig. 5(b)). An unstable system results in the robots diverging from the desired equilibrium, ultimately crashing into the net (Fig. 5(c)). A video of the experiments is available at http://mrsl.grasp.upenn.edu/nmichael/ISER2010.m4v.

The outcome of these initial experiments suggests that it is possible to define a communication rate and corresponding delay and compute the feasible gains for stability in the system while also modeling the control graph. We show that even for our simplified model there is a correspondance and the experimental results provide insight into the selection of appropriate gains given the control and network graphs and communication rate. Unexpectedly, in Fig. 2(b) we see that increasing the rate of communication may hurt more than help the performance of the system. In future experiments, we plan to pursue a closer model to our actual robots and address some fundamental questions, concerns, and limitations.

- Under what circumstances is it preferable to decrease the communication rate to improve stability?
- Is it possible to adjust the control gain and communication rate online while ensuring stability?
- Is performance improved by adopting a more specialized communication protocol such as that described in [5]?
- How do these results scale to larger teams of robots with many cycles?
- To what degree do these results change as the network graph diameter increases and information must flow over multiple hops?

References

1. N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Maintaining connectivity in mobile robot networks," in *Experimental Robotics: The Eleventh International Symposium*, ser. Springer Tracts in Advanced Robotics, O. Khatib, V. Kumar, and G. J. Pappas, Eds. Springer Berlin, 2009, vol. 54, pp. 117–126.
2. J. McLurkin, "Analysis and implementation of distributed algorithms for Multi-Robot systems," Ph.D. thesis, Massachusetts Institute of Technology, 2008.
3. H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 443–455, June 2004.
4. Y. Cai, K. Hua, and A. Phillips, "Leveraging 1-hop neighborhood knowledge for efficient flooding in wireless ad hoc networks," in *Proc. of the Int. Performance Computing and Communications Conf. (IPCCC)*, Phoenix, AZ, Apr. 2005, pp. 347–357.
5. B. J. Julian, M. Schwager, M. Angermann, and D. Rus, "A location-based algorithm for multi-hopping state estimates within a distributed robot team," in *Proc. of the Int. Conf. on Field and Service Robotics*, Cambridge, MA, July 2009.
6. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamical Systems*. Reading, MA: Addison-Wesley, 1994.
7. M. S. Brandicky, S. M. Phillips, and W. Zhang, "Stability of networked control systems: Explicit analysis of delay," in *Proc. of the American Control Conf.*, Chicago, IL, June 2000, pp. 2352–2357.
8. W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, Feb. 2001.
9. H. G. Tanner and D. K. Christodoulakis, "Decentralized cooperative control of heterogeneous vehicle groups," *Robotics and Autonomous Systems*, vol. 55, no. 11, pp. 811–823, November 2007.
10. R. Olfati-Saber and R. R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.
11. V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *Proc. of the Joint IEEE Conf. on Decision and Control and European Control Conf.*, Seville, Spain, Dec. 2005, pp. 2996–3000.
12. X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hendrick, "Effects of communication delay on string stability in vehicle platoons," in *Proc. of IEEE Intelligent Transportation Systems Conf.*, Oakland, CA, August 2001, pp. 625–630.
13. J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proc. of the IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.
14. M. Schwager, N. Michael, V. Kumar, and D. Rus, "Time scales and stability in networked multi-robot systems," in *Proc. of the Int. Workshop on the Algorithmic Foundations of Robotics*, Singapore, Dec. 2010, In Preparation.
15. N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP multiple micro UAV testbed," *IEEE Robotics and Automation Magazine*, May 2010, To Appear.
16. "Vicon Motion Systems, Inc." http://www.vicon.com.
17. "Ascending Technologies, GmbH," http://www.asctec.de.
18. "Robot Operating System (ROS)," http://www.ros.org.