International Journal of Control
Vol. 00, No. 00, Month 200x, 1–25

RESEARCH ARTICLE

Constrained Agreement Protocols for Tree Graph Topologies

Rahul Chipalkatty* and Magnus Egerstedt

Georgia Institute of Technology, Atlanta, GA 30332, USA;

(*February 2011*)

This paper presents a novel way of manipulating the initial conditions in the consensus equation such that a constrained agreement problem is solved across a distributed network of agents, particularly for a network represented by a tree graph. The presented method is applied to the problem of coordinating multiple pendula attached to mobile bases. The pendula should move in such a way that their motion is synchronized, which calls for a constrained optimal control problem for each pendulum as well as the constrained agreement problem across the network. Simulation results are presented that support the viability of the proposed approach as well as a hardware demonstration.

Keywords: consensus algorithm; optimal control; graph-based multi-agent control; constrained agreement protocol;

1 Introduction

The agreement protocol (or consensus equation) has by now emerged as a standard way in which to achieve agreement among agents in a distributed network. It can be utilized for anything from agreement in embedded physical systems like mobile robots or UAVs, to distributed computer networks, e.g. Tanner et al. (2000), Ren and Beard (2004), Dimarogonas and Kyriakopoulos (2007), Olfati-Saber and Murray (2003), Jadbabaie et al. (2003), Mesbahi and Egerstedt (2010). And, as the value on which the agents agree is dependent on the agents' initial conditions, it is not overly surprising that this dependency can be employed such that the final agreement state is guaranteed to satisfy certain constraints.

This work seeks to address the problem of finding an agreement state that satisfies a global state constraint given only local interactions among distributed agents. Specifically, within a group of agents, constraints exist between certain pairs of agents together with an information exchange, resulting in what is referred to in this paper as pairwise constraints. The pairwise constraints naturally lead to a network topology where nodes represent agents and edges represent pairwise constraints and information exchange.

The pairwise constraints that involve a particular agent are called that agent's local constraints and this particular agent only exchanges information with the agents with which it shares a pairwise constraint, hence the "local" descriptor. Subsequently, the global constraint for the entire group is the constraint that all agents satisfy their respective local constraints. Each agent can communicate its own state as well as its "opinion" of what the other agents' states should be (i.e. state opinions) such that the entire network's pairwise constraints are met.

As such, the goal of this work is to utilize the consensus equation to arrive at an agreement on the state opinions of every agent in the network such that every pairwise constraint in the network is satisfied (global constraint) given that only the local constraints are initially satisfied.

*Corresponding author. Email: chipalkatty@gatech.edu

ISSN: 0020-7179 print/ISSN 1366-5820 online

DOI: 10.1080/0020717YYxxxxxxxx
http://www.informaworld.com

We apply this constrained agreement protocol to the problem of controlling a collection of cart-driven pendula in a coordinated fashion. In particular, their mobile bases are to be controlled in such a way that, at some specified terminal time, they move in unison (with the same frequency and phase) at a fixed inter-pendula distance. This should be achieved using only local information, i.e. the control actions are only to be controlled based on information from neighboring pendula.

The problem is ensuring that the agreement protocol results in final state opinions for *all* pendula such that these final states satisfy the global constraint. Although running the standard consensus equation (e.g. Olfati-Saber and Murray (2003), Jadbabaie et al. (2003)) – or versions of the gossip algorithm Boyd et al. (2005), Xiao et al. (2005) – will result in an agreement, the agreed upon states are not guaranteed to satisfy the global constraint. However, in this work, we will demonstrate that the initial conditions for the agreement protocol can be manipulated such that the resulting state opinions will satisfy the global constraints.

Although optimal control techniques are used in the control formulation presented in this paper, the problem is, at heart, a distributed agreement problem rather than a distributed optimal control problem. For more on the latter problem see for example Rotkowitz and Lall (2006), Motee and Jadbabaie (2008), Bamieh et al. (2002) and Rantzer (2009), and the references therein. These works seek to solve distributive optimal control problems over a network while the work presented in this paper simply utilizes optimal control to drive the system to a desired state as determined through an agreement protocol. Any form of control could, in principle, have been used here. However, optimal control provides an effective method for solving the types of constrained problems under consideration in this paper.

Relevant work on agreement for systems with constraints or oscillating dynamics include Jadbabaie et al. (2004), where the stability of the Kuromoto model of coupled nonlinear oscillators was investigated, and Moore and Lucarelli (2005), Nedic et al. (2008), where constrained consensus was considered. However, in the former case, no constraints were present, and in the latter case, the proposed solution required that the consensus update law be modified over time with time-varying weights. In contrast to this, Chipalkatty et al. (2009) presents a simple, static update law for achieving agreement while satisfying the constraints for a line topology network. In this paper, we will present a novel method of using a static update law to satisfy global constraints over not just a line topology, but any arbitrary directed tree graph topology. We will also present the viability of this approach though a hardware demonstration

The outline of this paper is as follows: In Section 2, we introduce the distributed network of agents and its constraints, followed by a discussion in Section 3 about the constrained consensus problem. In Section 4, we show how selecting the initial conditions appropriately leads to an agreement state that satisfies the global constraints and in Sections 5 and 6, we show how this method is applied to the networked pendulum example. Simulation results are presented in both Sections 5 and 6 as well as hardware results in Section 5.

2 Problem Formulation

2.1 *Notation*

A graph, G, is defined by a node set, $V = \{1, 2, 3, \dots, N\}$ of N nodes and an edge set $E \subset V \times V$ of unordered node pairs. Two nodes, i and j, are adjacent, or neighbors, if $(i, j) \in E$. The neighborhood set of a node $i \in V$, $\mathbb{N}_i$, is the set of nodes $j \in V$ adjacent to node i. A tree graph is a specific type of graph where there exists only one path from a particular node to any other node in the graph. Note that for tree graphs, the number of edges, M, in the graph is given by $M = N - 1$. Edges can be given an orientation using $\sigma : E \to \{-1, 1\}$, resulting in a directed graph, G^σ, for which an associated incidence matrix, $D = [D_{ij}] \in \mathbb{R}^{N \times M}$, has elements given by

$$D_{ij} = \begin{cases} 1 & \text{if vertex } i \text{ is the tail of edge } j \\ -1 & \text{if vertex } i \text{ is the head of edge } j \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

for $i = 1 \dots N$ and $j = 1 \dots M$.

2.2 *Multi-Agent Network*

The multi-agent network in this paper is modeled by a tree graph, $G = (V, E)$ where the N nodes correspond to N agents. The M edges of the network correspond to information exchange between agents as well as a pairwise constraint. Each edge is assigned an arbitrary orientation, as given by σ, resulting in an associated incidence matrix, D. Note, however, that the graph is still an undirected one, so the information exchange is undirected. Representing the network through an incidence matrix is key here because the incidence matrix will be utilized to construct formal definitions of the local and global constraints.

In order to illustrate the operations used later, an example network is presented and the prescibed operations will be performed on this network throughout the paper.

Example Graph:

The example tree graph, $\hat{G} = (\hat{V}, \hat{E})$ with $\hat{V} = \{1, 2, 3, 4, 5\}$ and $\hat{E} = \{(1, 2), (1, 4), (4, 3), (4, 5)\}$, is given with arbitrary edge orientations as presented in Figure (1). The corresponding incidence matrix, $\hat{D}$, is

$$\hat{D} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{2}$$

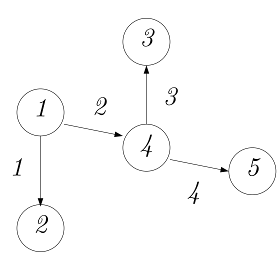


Figure 1. Tree Graph example, $\hat{G}$

2.3 *States and Opinions*

Each agent in the network has an associated state and we let $x_{ii} \in \mathbb{R}^l$ denote agent i's state for all $i \in V$. The double indices are used here to denote agent i's opinion of itself, which coincides with its state, and this notation is now used again to denote an agent's opinions of other agents.

Each agent in the network will form a state opinion of all the other agents in the network and share those opinions with its neighbors. Let $x_{ij}(t) \in \mathbb{R}^l$ be agent i's opinion of what agent j's state should be such that all the pairwise constraints in the network are satisfied (i.e agent

i's' state opinion of agent j). The collection of these opinions is denoted by the state opinion vector, x_i, for each agent, where $x_i = [x_{i1}, \ldots, x_{ii}, \ldots x_{iN}]$ for all $i \in V$. A form of this vector will eventually become the quantity used in the proposed agreement protocol.

2.4 *Pairwise Constraints*

Let each edge with arbitrary orientation denote an associated linear pairwise constraint on agents i and k for all nodes $i, k \in V$ where $(i, k) \in E$. This linear pairwise constraint between agents is defined as

$$P(x_{ii} - x_{kk}) = b, \tag{3}$$

with vertex i being the tail of the edge, $b \in \mathbb{R}^p$, and $P \in \mathbb{R}^{p \times l}$. Note that P has full row rank. This particular form for the pairwise constraints was chosen because it is linear with respect to the difference in the agent's states, allowing for relative constraints, e.g. for synchronizing movement between the agents.

2.5 *Constraint Definition*

Given a pairwise constraint for every edge, the global constraint for the network is defined as the collection of all pairwise constraints and is satisfied when the state of every node in the network satisfies all pairwise constraints. An agent's local constraints are defined as the set of all pairwise constraints it shares with its neighbors. When a particular node's state satisfies all the pairwise constraints with its neighbors, this agent's state satisfies its local constraints.

The following is a discussion on how to construct the local constraints for each node and the global constraints for the network given the graph, G, and the incidence matrix, D. Since the incidence matrix gives us a represention of all the edges in the network, the incidence matrix will first be utilized to construct an expression for the global constraint on the network.

Let $x_g \in \mathbb{R}^{lN}$ be a state opinion vector such that all it's state opinions satisfy all the pairwise constraints in the network. Then, the global constraint is defined as

$$(D^T \otimes P)x_g = \mathbf{1} \otimes b, \tag{4}$$

where $\otimes$ denotes the Kroenecker product and $\mathbf{1}$ is a M dimensional vector with all entries being 1.

From the example network G, the global constraint, $(D^T \otimes P)x_g = \mathbf{1} \otimes b$, is satisfied if, for $x_g = [x_{11}, \ldots, x_{55}]^T$,

$$\begin{bmatrix} P & -P & 0 & 0 & 0 \\ P & 0 & 0 & -P & 0 \\ 0 & 0 & -P & P & 0 \\ 0 & 0 & 0 & P & -P \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{22} \\ x_{33} \\ x_{44} \\ x_{55} \end{bmatrix} = \begin{bmatrix} b \\ b \\ b \\ b \end{bmatrix}. \tag{5}$$

Here, since each row corresponds to an edge, this expression represents every pairwise constraint in the network. The following operations isolate the rows of the transposed incidence matrix that correspond to edges that node i is incident with and apply the pairwise constraints to these edges to arrive at an expression for the local constraints of agent i.

For all $i \in V$, we define the diagonal matrix, $F^i \in \mathbb{R}^{M \times M}$, whose mth diagonal element is 1 if $D_{im} \neq 0$ and 0 otherwise, as well as the complement, $F^{ic} \in \mathbb{R}^{M \times M}$, which is a diagonal matrix whose mth diagonal element is 0 if $D_{im} \neq 0$ and 1 otherwise, for $i = 1 \ldots N$ and $m = 1 \ldots M$. To illustrate this further, the F^i and F^{ic} matrices are shown for agent 4 for the example graph

$\hat{G}$ as

$$\hat{F}^4 = \begin{bmatrix} 0\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix}, \; \hat{F}^{4c} = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \end{bmatrix}.$$

Note $\hat{F}^4$ contains a 1 in the column or row corresponding to the edges agent 4 is incident with and vice versa for $\hat{F}^{4c}$

Then, let $T_i \in \mathbb{R}^{M \times N}$ be defined as

$$T_i = F^i D^T$$

with the compliment, T_i^c, being

$$T_i^c = F^{ic} D^T.$$

Note that $F^i + F^{ic} = I$, where I is the $M \times M$ identity matrix. Again, the example is worked further to show the matrices resulting from these operations. T_i and T_i^c for agent 4 are

$$\hat{T}_4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix},$$

$$\hat{T}_4^c = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Lemma 2.1: *Given matrices T_i and T_i^c for every agent i in the network for $i = 1, \ldots, N$, then $T_i^c + T_i = D^T$*

Proof

$$\begin{aligned} T_i^c + T_i &= F^i D^T + F^{ic} D^T = \\ &= (F^i + F^{ic}) D^T = I D^T = D^T \end{aligned}$$

□

Here, the rows of $T_i \in \mathbb{R}^{M \times N}$ encode only the edges that node i is incident with while the remaining rows have all elements as zero. Similarly, the rows of T_i^c encode the edges that vertex i is not incident with and edges incident with node i have all zero elements.

Additionally, a vector is needed to equate the appropriate constraints in each row to b. In other words, each non-zero row in $(T_i \otimes P)x$ must equal b. The vector, $f^i = [f_m^i]_{m=1}^M \in \mathbb{R}^M$, is defined as having its mth element be 1 if $T_{im} \neq 0$ or 0 otherwise, for $i = 1 \ldots N$ and $m = 1 \ldots M$. Continuing with our example network, $\hat{f}_4 = [0\ 1\ 1\ 1]^T$.

Lemma 2.2: *Given the vector, f^i, for every agent i in the network for $i = 1, \ldots, N$, then*

$$\sum_{i=1}^{N} f^i = \mathbf{21}.$$

Proof For every edge j in the network, there are two agents incident with that edge. Therefore, a 1 will appear in the jth element for two agents, resulting in a 2 in every element of the sum of f^i's. □

The matrix, T_i, can now be used to define the local constraints on agent i. Let x_i satisfy pairwise constraints that involve agent i, such that

$$(T_i \otimes P)x_i = f^i \otimes b, \tag{6}$$

for $i = 1 \ldots N$. In other words, only the pairwise constraints agent i can satisfy are represented in (6). The local constraints for agent 4 in the graph example $\hat{G}$, would be satisfied if, for $x_4 = [x_{41}, \ldots, x_{45}]^T$, $(\hat{T}_4 \otimes P)x_4 = \hat{f}^4 \otimes b$ or

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ P & 0 & 0 & -P & 0 \\ 0 & 0 & -P & P & 0 \\ 0 & 0 & 0 & P & -P \end{bmatrix} \begin{bmatrix} x_{41} \\ x_{42} \\ x_{43} \\ x_{44} \\ x_{45} \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ b \\ b \end{bmatrix}.$$

Note that in the example, agents 2 and 4 are not neighbors so agent 4 has no basis on which value to assign to x_{42}. Also note that this value is not required for agent 4's local constraints to be satisfied. However, in order to utilize the consensus equation , these agents must assign state opinions to non-adjacent agents.

In summary, each agent assigns a state opinion value to neighbors based on the initial state values communicated such that they meet the local constraints, (6). Then, the consensus equation will be used to reach an agreement on what the state opinion vectors should be. This agreed upon state opinion vector needs to satisfy the global constraint, (4). So the problem, here, is to determine how each agent should initialize their state opinion vectors so that the global constraint is satisfied by the result of the consensus equation.

3 Applying the Consensus Equation

Again, the goal of this work is to find a state opinion vector that satisfies the global constraint among distributed agents using only local interactions. In order to accomplish this, the consensus algorithm will be used to have the agents come to an agreement on such a state. It should be noted that each agent's initial state opinions will satisfy the local constraints for that agent only.

Recall x_i is the quantity over which the consensus equation will be run and it is desired that the resulting quantity satisfies the global constraint. For agents in the neighborhood of agent i, the initial state opinions are made based on the states of the agent as communicated by those agents. However, for agents not in the neighbodhood of agent i, state opinions are still required to construct the state opinions vector and utilize the consensus equation. However, the lack of communication means that initializing the state opinion vector with state opinions for non-adjacent agents is a problem. The following discusses how to deal with this problem by first simply setting these values to zero and showing why the resulting agreement state does not satisfy the global constraint.

3.1 *Attempt 1: Filling the state opinion vector with* **0***'s*

For undirected connected graphs, the consensus equation, as given by

$$\dot{x}_i(t) = -\sum_{k \varepsilon \mathbb{N}_i} (x_i(t) - x_k(t)) \ \text{ for } i = 1 \ldots N, \tag{7}$$

converges to the invariant centroid, $x_c \in \mathbb{R}^{Nl}$, where

$$x_c = \frac{1}{N}\sum_{i=1}^{N} x_i(0) \tag{8}$$

(See, for example, **?**) Note that the consensus algorithm requires that agents communicate their state opinion vectors with their neighbors and vice versa.

Suppose the initial state opinions of non-adjacent agents are simply set to zero. Using the example graph, it will be shown that the resulting agreement state does not satisfy the global constraint although the initial conditions of each agent's state opinion vector satisfy that agent's local constraints. Let the initial state opinion vectors be

$$x_1 = \begin{bmatrix} x_{11} \\ x_{12} \\ 0 \\ x_{14} \\ 0 \end{bmatrix}, x_2 = \begin{bmatrix} x_{21} \\ x_{22} \\ 0 \\ 0 \\ 0 \end{bmatrix}, x_3 = \begin{bmatrix} 0 \\ 0 \\ x_{33} \\ x_{34} \\ 0 \end{bmatrix},$$

$$x_4 = \begin{bmatrix} x_{41} \\ 0 \\ x_{43} \\ x_{44} \\ x_{45} \end{bmatrix}, x_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ x_{54} \\ x_{55} \end{bmatrix}.$$

So, the resulting agreement state is

$$x_c = \begin{bmatrix} \frac{x_{11}+x_{21}+x_{41}}{5} \\ \frac{x_{12}+x_{22}}{5} \\ \frac{x_{33}+x_{43}}{5} \\ \frac{x_{14}+x_{34}+x_{44}+x_{54}}{5} \\ \frac{x_{45}+x_{55}}{5} \end{bmatrix}. \tag{9}$$

Given that the inital state opinion vectors satisfy the local constraints,

$$(\hat{T}_1 \otimes P)x_1 = \hat{f}_1 \otimes b = \begin{bmatrix} Px_{11} - Px_{12} \\ Px_{11} - Px_{14} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} b \\ b \\ 0 \\ 0 \end{bmatrix}, \tag{10}$$

$$(\hat{T}_2 \otimes P)x_2 = \hat{f}_2 \otimes b = \begin{bmatrix} Px_{21} - Px_{22} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{11}$$

$$(\hat{T}_3 \otimes P)x_3 = \hat{f}_3 \otimes b = \begin{bmatrix} 0 \\ 0 \\ Px_{34} - Px_{33} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b \\ 0 \end{bmatrix}, \tag{12}$$

$$(\hat{T}_4 \otimes P)x_4 = \hat{f}_4 \otimes b = \begin{bmatrix} 0 \\ Px_{41} - Px_{44} \\ Px_{44} - Px_{43} \\ Px_{44} - Px_{45} \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ b \\ b \end{bmatrix}, \tag{13}$$

and

$$(\hat{T}_5 \otimes P)x_5 = \hat{f}_5 \otimes b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ Px_{54} - Px_{55} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ b \end{bmatrix}. \tag{14}$$

The result of plugging values from the local constraints into the global constraint is that

$$(\hat{D}^T \otimes P)x_c = \frac{2}{5} \begin{bmatrix} b + \frac{1}{2}Px_{41} \\ b + \frac{1}{2}P(x_{21} - x_{34} - x_{54}) \\ b + \frac{1}{2}P(x_{14} + x_{54}) \\ b + \frac{1}{2}P(x_{14} + x_{34}) \end{bmatrix} \neq \begin{bmatrix} b \\ b \\ b \\ b \end{bmatrix} \tag{15}$$

and therefore the agreement state does not satisfy the global constraint.

As a result, it is proposed that local information (i.e the state opinion vector shared by an agent's neighbors) be used to initialize the state opinions for non-adjacent agents such that the extra terms in each row of (15) will be canceled, i.e. x_{41} in row 1, x_{21}, x_{34}, x_{54} in row 2, x_{14}, x_{54} in row 3 and x_{14}, x_{34} in row 4. In addition, a gain value will be needed to cancel the $\frac{2}{5}$ term in (15). By doing this, it will be shown that the resulting agreement state satisfies the global constraint.

3.2 *Attempt 2: Initial Condition Definition by Propogation*

It is proposed that the state opinions of agents not adjacent to agent i should be assigned values in x_i such that the equation

$$(T_i^c \otimes P)x_i = \mathbf{0}_M \tag{16}$$

is satisfied, where $\mathbf{0}_M$ is a M dimensional vector with all elements being zero. Note that this requires that each agent a priori knows the structure of the network in the form of the incidence matrix.

For each agent, this equation is sufficient to choose state opinion values for every non-adjacent agent. For any agent i, there are d_i agents adjacent to agent i. Therefore, there are $N - 1 - d_i$ agents not adjacent to agent i, which implies $(N - 1 - d_i)l$ state opinions are undefined. T_i^c contains $(M - d_i)l$ rows that represent pairwise constraints. This results in $(M - d_i)l$ equations and $(N - 1 - d_i)l$ unknowns.

As a result, effectively choosing initial state opinions is feasible only if $M = N - 1$, i.e. G is a tree graph. For graphs with cycles, $M > N - 1$, therefore, the system of equations is overdetermined. For incomplete graphs, $M < N - 1$, resulting in a system of equations that is insufficient to solve for all the initial state opinions.

To give further insight on the resulting state opinion assignments, we will return to the example graph, G. To continue the example for agent 4, the intermediate matrix T_4^c results in

$$T_4^c = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

So, we assign non-adjacent agents state opinions according to

$$(T_4^c \otimes P)x_4 = \begin{bmatrix} P(x_{41} - x_{42}) \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

x_{41} is a communicated value because agent 1 is adjacent to agent 4, so x_{42} is initially assigned the same value as x_{41}, i.e. it is assigned the value of the agent adjacent to agent 4 that is in the path from agent 2 to agent 4. Similarly for agent 5,

$$(T_5^c \otimes P)x_5 = \begin{bmatrix} P(x_{51} - x_{52}) \\ P(x_{51} - x_{54}) \\ P(x_{54} - x_{53}) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

reveals that as x_{54} is defined through communication, x_{51} and x_{53} are set to x_{54}. As a result, x_{52} is also set to the value of x_{54}. Figures 2 and 3 show this graphically for agents 4 and 5. Nodes of the same color denote that the initial state opinions are equal from the view of the red colored node.

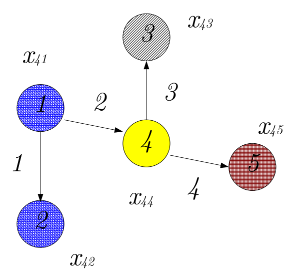


Figure 2. Initial conditions propogation for example graph agent 4. The state opinion for non-adjacent agent 2 is assigned the same value as adjacent agent 1.

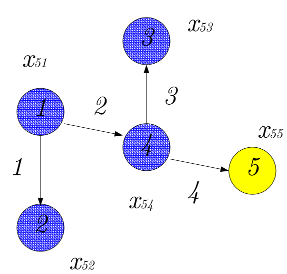


Figure 3. Initial conditions propogation for example graph agent 5. The state opinion for non-adjacent agents 1, 2, and 3 are assigned the value of adjacent agent 4.

In other words, the state opinion assignments propagate out into the network from the neighborhood of agent i as shown in Figure4. By setting the state opinion of a non-adjacent agent j to the same value an adjacent agent on the path from agent i to the non-adjacent agent j, we are guaranteeing that the pairwise constraints on that agent results in $\mathbf{0}_l$ in agent i's opinion.

As a result, each element of x_i is defined and it will now be shown that this choice of initial conditions for the consensus equation will result in an agreement state that satisfies the global constraint. The quantity used in the consensus equation is now denoted, $X_i(0) = \gamma x_i$ for $i =$

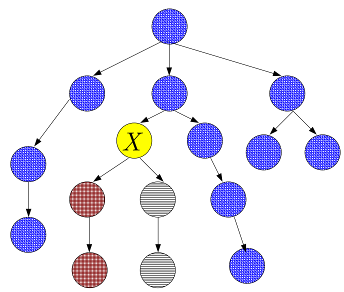


Figure 4. Directed Tree Graph example for initial conditions propagation. The figure represents the view of the graph from the perspective of agent X. Agent X assigns state opinions for non-adjacent agents according to the colors shown in this diagram: in agents X's state opinion vector, values of non-adjacent agents are set to values of agents adjacent to agent X that are in the path from the non-adjacent agent to agent X.

$1 \dots N$ with $X_i \in \mathbb{R}^{Nl}$ and gain $\gamma \in \mathbb{R}$. The consensus equation,

$$\dot{X}_i(t) = -\sum_{k \varepsilon \mathbb{N}_i} (X_i(t) - X_k(t)) \text{ for } i = 1 \dots N \tag{17}$$

results in

$$X_c = \frac{1}{N} \sum_{i=1}^{N} X_i(0) = \frac{\gamma}{N} \sum_{i=1}^{N} x_i \tag{18}$$

as the final agreement state, $X_c \in \mathbb{R}^{Nl}$.

It is required that this agreement state satisfy the global constraint where each agent's state opinion vector satisfies the pairwise constraint for each edge in the graph. Therefore, it is needed that

$$(D^T \otimes P) X_c = \mathbf{1} \otimes b. \tag{19}$$

To verify that the agreement state satisfies the global constraint, we can plug in the RHS of (18) for X_c and rewrite the global constraint, (19) as

$$\begin{aligned} (D^T \otimes P) X_c &= (D^T \otimes P) \frac{\gamma}{N} \sum_{i=1}^{N} x_i \\ &= \frac{\gamma}{N} \sum_{i=1}^{N} (D^T \otimes P) x_i. \end{aligned} \tag{20}$$

Using (6), (20) can be written as

$$= \frac{\gamma}{N} \sum_{i=1}^{N} ((T_i + T_i^c) \otimes P) x_i. \tag{21}$$

By the distributive property of the Kronecker product and (6), (21) becomes

$$= \frac{\gamma}{N}\sum_{i=1}^{N}(T_i \otimes P)x_i + \frac{\gamma}{N}\sum_{i=1}^{N}(T_i^c \otimes P)x_i$$
$$= \frac{\gamma}{N}\sum_{i=1}^{N} f^i \otimes b + \frac{\gamma}{N}\sum_{i=1}^{N}(T_i^c \otimes P)x_i. \tag{22}$$

Recalling (6), (22) is rewritten as

$$= \frac{\gamma}{N}2\mathbf{1} \otimes b + \frac{\gamma}{N}\sum_{i=1}^{N}(T_i^c \otimes P)x_i. \tag{23}$$

Plugging (16) into (23) results in

$$(D^T \otimes P)X_c = \frac{\gamma}{N}2\mathbf{1} \otimes b$$

Plugging in $\gamma = \frac{N}{2}$, it is shown that

$$(D^T \otimes P)X_c = \mathbf{1} \otimes b.$$

Therefore, the global constraint is satisfied by the final agreement state through the manipulation of the initial conditions of the consensus equation. This leads us to the following theorem.

Theorem 3.1 : *If the state opinions of agents not adjacent to agent i in X_i are assigned values such that $(T_i^c \otimes P)x_i = 0_M$ and $\gamma = \frac{N}{2}$, then the resulting agreement state satisfies the global constraint, $(D^T \otimes P)X_c = \mathbf{1} \otimes b$.*

4 2D Pendulum Dynamics and Control

To investigate the viability of this approach, we apply the algorithm to the synchronization of a distributed network of planar mass-cast pendula. The planar dynamics of the system allows us to analyze a network that is modeled by a simple form of the tree graph, a line graph.

4.1 *Dynamics*

The dynamics of a single cart-pendulum system (referred to as an agent) can be derived using Lagrange's Equations (e.g. Spong (1989))

$$\frac{d}{dt}\left(\frac{\partial \mathfrak{L}}{\partial \dot{q}}\right) - \frac{\partial \mathfrak{L}}{\partial q} = Q,\ \mathfrak{L} = K - T,$$

where K is the kinetic energy of the system, T is the potential energy of the pendulum, Q is the parameterized forces acting on the system, and $\mathfrak{L}$ is the Lagrangian, where q is the vector of parameters, $[v\ \theta]^T$ as shown in Figure 5.

We can define the kinetic and potential energy as well as the parameterized forces acting on the system. Based on Figure 5, the only parameterized force, Q, on the system is the force, F, applied in the v direction. This force will be the control input, u, to the system. No damping force is considered in this model as pendula can be approximated as zero damping systems. The

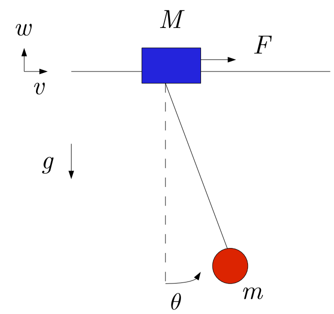


Figure 5. Pendulum Diagram

resulting equations of motion are

$$\begin{aligned}\ddot{\theta} =& -\frac{\ddot{v}}{l}cos(\theta) - \frac{g}{l}sin(\theta)\\ \ddot{v} =& -\frac{ml\ddot{\theta}}{M+m}cos(\theta) + \frac{ml\dot{\theta}^2}{M+m}sin(\theta) + \frac{u}{M+m}.\end{aligned}$$

4.2 *Linearization*

These dynamics can be linearized about the $\theta = 0,\ \dot{\theta} = 0,\ \dot{v} = 0$ equilibrium point. Here, we denote v_i and θ_i as the velocity and angle associated with the ith pendulum. As such, the single pendulum system for the ith pendulum is given as $\dot{x}_{ii}(t) = A_i x_{ii}(t) + B_i u_i(t)$, where:

$$x_{ii} = \begin{bmatrix} v_i \\ \dot{v}_i \\ \theta_i \\ \dot{\theta}_i \end{bmatrix}, A_i = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-(M+m)g}{Ml} & 0 \end{bmatrix}, B_i = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{-1}{Ml} \end{bmatrix}.$$

Note that this pair, (A_i, B_i), is completely controllable.

For a N planar pendulum system, the system can be written as $\dot{x}(t) = Ax(t) + Bu(t)$, where

$$\begin{aligned} x^T =& \left[x_{11}^T, \ldots, x_{NN}^T\right], u = \left[u_1 \ldots u_N\right]'^T \\ A =& \begin{bmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_N \end{bmatrix}, B = \begin{bmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_N \end{bmatrix}.\end{aligned}$$

Note that in this paper, $A_i = A_j$ and $B_i = B_j$ for $i, j = 1, \ldots, N$, since the pendula are assumed to be homogeneous.

4.3 *Assumptions*

Throughout this section, some assumptions are made and here we gather the assumptions for the sake of easy reference. We first assume that each cart-pendulum system can measure its own cart position, cart velocity, pendulum angle, and pendulum angular velocity. It is also assumed that pendulum angles and angular velocities are small enough so that the linearized dynamics can be used to model the system behavior. Damping is also assumed to be small enough to approximate it as exerting zero forces on the system. It is also assumed adjacent agents can communicate state opinion vectors with each other.

4.4 *Planar Pendula Network*

Since the pendula lie in a plane, the natural resulting network topology is a line graph as in Figure 6.

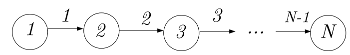


Figure 6. N Pendula Line Graph

The corresponding pairwise constraint is that the adjacent pendula achieve synchronization or, more specifically, that they achieve identical angles, angular velocities, and cart velocities while maintaining a set distance, d, between the carts. In terms of the pendula states, it is required that $v_i - v_j = d$, $\dot{v}_i - \dot{v}_j = 0$, $\theta_i - \theta_j = 0$, $\dot{\theta}_i - \dot{\theta}_j = 0$, i.e. $P(x_i - x_j) = b$ where $P = I_4$ and $b = [d\ 0\ 0\ 0]^T$.

With the network topology and pairwise constraints defined, each agent must define its state opinion vector. In addition, the global constraint can be defined using (4) and it is clear that we wish to drive the system to a state that satisfies this constraint. What must now be defined is how the system will be driven to such a state, once it is found, and how the state opinion vectors will be initialized for both adjacent and non-adjacent agents. The following constrained optimal control problem will give us both a way to initialize the state opinion vectors for each agent as well as a control law to drive each pendula to the agreement state that results from the consensus equation.

4.5 *Optimal Control*

For the following control law derivation, we treat a collection of pairwise constraints as a terminal constraint on the linearized N agent system. This linear state constraint is denoted by $Cx(T) = k$.

The associated minimum energy, point-to-point transfer control problem with terminal time, T, becomes

$$\min_u J(u(t)) = \int_0^T ||u(t)||^2 dt \tag{24}$$

such that

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ x(0) &= x_0 \\ x(T) &= x_T, \end{aligned}$$

where we assume that x_T satisfies the constraints, i.e. $Cx_T = k$. The solution to this common optimal control problem is

$$u_{opt}(x_T) = B^T e^{A^T(T-t)} W^{-1}(x_T - e^{AT} x_0),$$

where the Grammiam, W, is invertible and positive definite due to the controllability of the system. Plugging $u_{opt}(x_T)$ back into the cost gives
$J(u_{opt}(x_T)) =$

$$= (e^{AT}x_0 - x_T)^T e^{-A^T T} W^{-1} e^{-AT}(e^{AT}x_0 - x_T).$$

Since x_T is not unique, the goal now is to find the x_T that minimizes (25), which can be formulated as a quadratic programming problem,

$$\min_{x_T} \frac{1}{2} x_T^T Q x_T + R x_T$$

such that $Cx_T = k$, where

$$Q = 2e^{-A^T T} W^{-1} e^{-AT}$$
$$R = -2x_0^T W^{-1} e^{-AT}.$$

The unique solution,

$$x_{T_{opt}} = Q^{-1}(-R^T + C^T(CQ^{-1}C^T)^{-1}(k + CQ^{-1}R^T), \tag{25}$$

gives the control,

$$u_{opt}(x_{T_{opt}}) = B^T e^{A^T(T-t)} W^{-1}(x_{T_{opt}} - e^{AT}x_0). \tag{26}$$

The enabling observation now is that we can use (25) to initialize the state opinions of adjacent agents as to where they should be driven such that the local constraints are satisfied and subsequently use these state opinions to initialize state opinions of non-adjacent agents. Then, the consensus equation can be utilized to update each pendulum's state opinion vectors. As shown earlier, a final state opinion vector will be found that satisfies the global constraint for the entire network and the control law, (26), can be used to drive the system to this state, synchronizing the pendula.

4.6 *System Control: Consensus Algorithm + Optimal Control*

The following summarizes the algorithm used. First, each agent communicates its current state to its neighbors. Then, using these states, each agent calculates its initial state opinion for its neighbors using (25). Following this, each agent then assigns state opinions to non-adjacent agents using (**??**). These values are now collected in a state opinion vector for each agent. Next, each agent will communicate its state opinion vector at each time instant and use the consensus equation to update its state opinion vector. Finally, the state opinion vector is then used in (26) to calculate a control input. The state opinion vector update and control input calculation are repeated at every time instant until the terminal time, T.

Note that this implies that the consensus equation converges prior to the terminal time of the optimal control problem. For example, for an agent i not on the boundary(i.e. not agent 1 or N), the agent recieves the state of agent $i-1$ and agent $i+1$ and creates the vector $x(0) = [x_{i-1}\ x_i\ x_{i+1}]^T$ with

$$C = \begin{bmatrix} P & -P & 0 \\ 0 & P & -P \end{bmatrix} \tag{27}$$

and $k = [b^T \; b^T]^T$. These quantities are then used in (25) to initialize the state opinions of the agents two neighbors. After updating the state opinion vector through the consensus equation, agent i uses the values $X_i(t)$ to calculate its control at time t, for $i = 1, \ldots, N$. Let $X_{ii}(t)$ be the ith element of $X_i(t)$ such that the following control law is calculated,

$$u_i(X_{ii}(t)) = B_i^T e^{A_i^T(T-t)} W_i(t)^{-1} (X_{ii}(t) - e^{A_i T} x_{ii}(t)). \tag{28}$$

Recall x_{ii} is the current state of agent i and A_i and B_i are the corresponding single pendulum state space model. $W_i(t)$ is the Grammian for the pair (A_i, B_i) from time t to T. We now have a control that drives the entire network to a terminal state that satisfies the global constraint.

4.7 *Simulation Results*

The stated control laws are implemented in a MATLAB simulation of the presented pendulum dynamics. Simulations are run with the following parameters: $g = 9.8\ m/s,\ l = 0.30\ m,\ M = 1\ kg,\ m = 0.2\ kg$, and $d = 1.0\ m$ for the pendulum model. It should be noted that in order for this distributed control strategy to be effective, the consensus algorithm must converge to the agreement value before the specified final time in the optimal control law, which in this case is 20 seconds.

In Figure 7, the results are shown for a five pendula scenario using the optimal control law and the consensus algorithm. As a comparision, the same initial conditions are run for the centralized case, where full network state information is known to all agents, i.e. only the optimal control law is needed without the consensus equation. The results of this case are shown in Figure 8.

It can be seen for both cases that at 20 seconds, the distance between adjacent pendula is close to 1 m, as prescribed, while the velocities, angles, and angular velocities are identical for all the pendula. The animations of these scenarios are given in Figure 9.

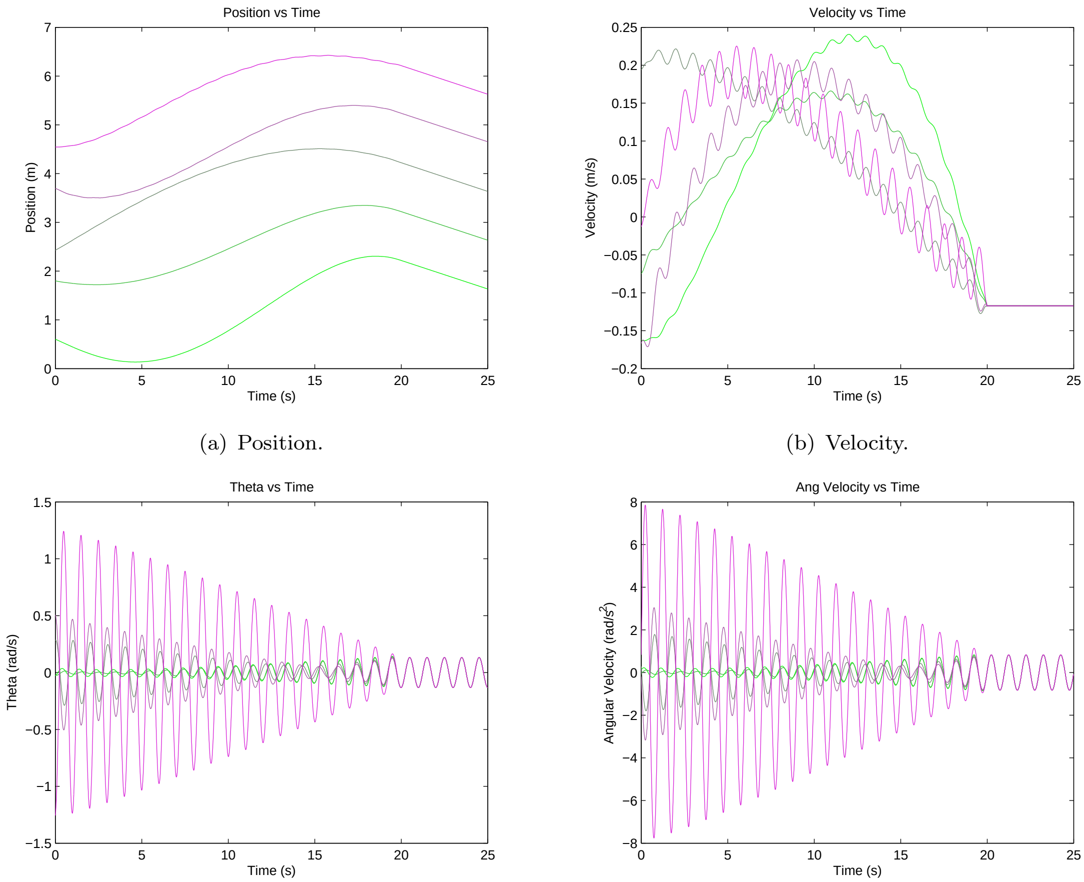
4.8 *Experimental Results*

Simulations of the proposed decentralized control system were deemed successful, so to further show the viability of this approach, a physical hardware demonstration was built. The testbed consists of three planar mass cart pendula placed on three different parallel tracks as shown in Figure 12(b). The goal is to synchronize the pendula swinging with zero cart position and velocity after 10 seconds starting from an unsychronized initial condition. The representative graph of the communication exchange and constraint directions is a 3 node line tree graph in the same configuration as Figure 6 with $N = 3$.

The demonstration structure and mass cart-pendula was constructed using Lego blocks as they serve as a effective rapid prototyping platform. Aluminum tubing was used as a track for the mass-cart pendula to slide upon and a Vex Sprocket chain was used to propel the carts. The effective pendulum length was 32.8 cm. The chain sprockets were driven by AX12+ Servo motors as part of the Robotis Bioloid Robotics Kit. The Servo motors are connected to a CM-5 micro-controller which receives commands over a serial communication port from a PC. The control software is implemented on the PC in a Java programing environment.

The AX12+ servo motors are equipped with integrated encoders relaying position and velocity sensor information at 0.10 second update rate with a 0.0051 radian resolution. A Nubotrics WW-02 WheelWatcher optical encoder was mounted to each pendulum to sense pendulum angle and angular velocity. Data was updated at a 0.1 sec rate to match the AX12+ sensor rate at a 0.049 radian resolution. This particular sensor was chosen for its floating optical wheel that minimizes friction.

The system hardware/software architecture is shown in Figure 10. Motor commands are sent out as velocity commands as the motors cannot be controlled by force commands. Software for the CM-5 is written in C code. Sensor data is passed through a 4th order FIR low-pass filter



(a) Position. (b) Velocity.

(c) Angle. (d) Angular Velocity.

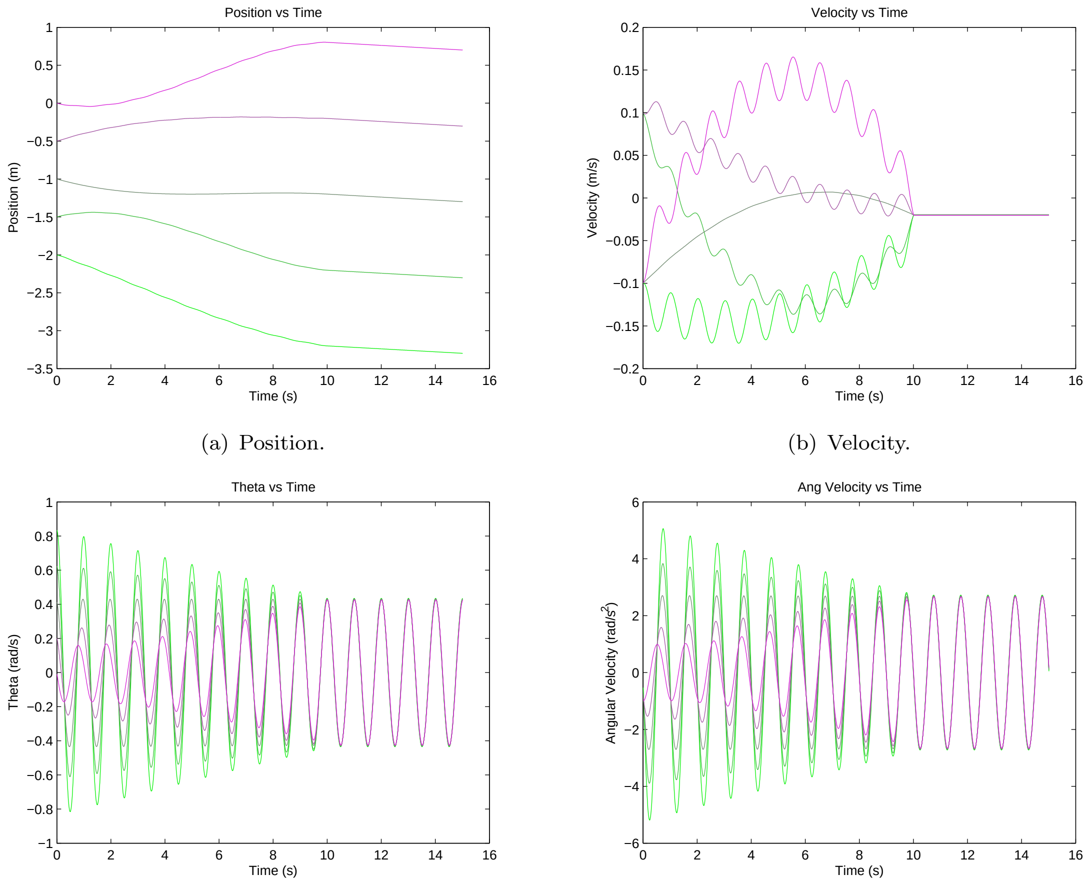
Figure 7. Synchronization control of 5 distributed pendula results showing final positions are 1m apart with identical final velocity, angle, and angular velocity.

designed in MATLAB.

The initial conditions for the demonstration run as seen in the attached video were created by running each pendulum at a sinusoid function of different phase shifts at the natural frequency of the pendulum, 0.8688 Hz , for 5 sec. Then, the velocity control commands are then applied for 10 sec. As shown in Figures 11, at the end of the 10 sec. period, the pendulum angles and anglular velocities are within 0.049 radians of each other implying the pendulum swinging is synchronized within that error. The data in the plots are smoothed using a 4th order spline in MATLAB and the 0.049 radian angle resolution error is compensated for in the pendulum angle data.

5 3D Pendulum Dynamics and Optimal Control

We, now, want to show the versatility of this technique by applying it to a network system modeled by any arbitrary tree graph. The application is extended to the synchonization of a distributed network of spherical pendula. We develop the application further by defing the dynamics, control, and consensus problem associated with this example. The main differences from the planar example are the network topology and pendulum dynamics. The consensus algorithm and control laws are similar to the planar pendula case.



(a) Position.

(b) Velocity.

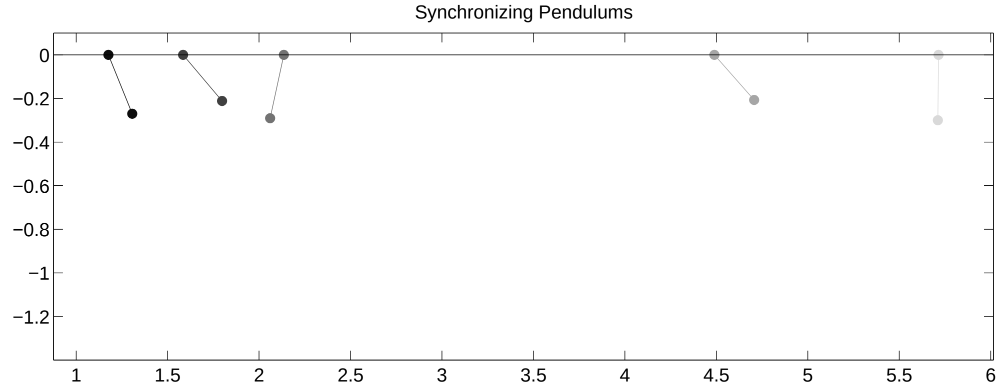
(c) Angle.

(d) Angular Velocity.

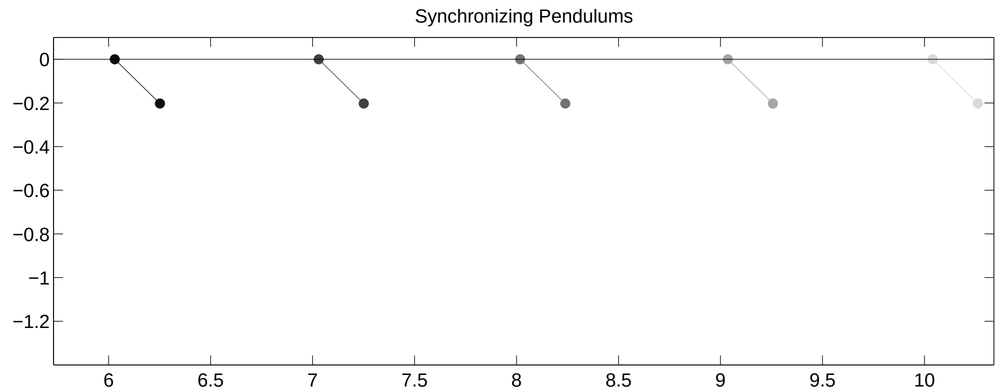
Figure 8. Synchronization control of 5 centralized pendula results showing final positions are 1m apart with identical final velocity, angle, and angular velocity.

5.1 *3D Pendulum Dynamics*

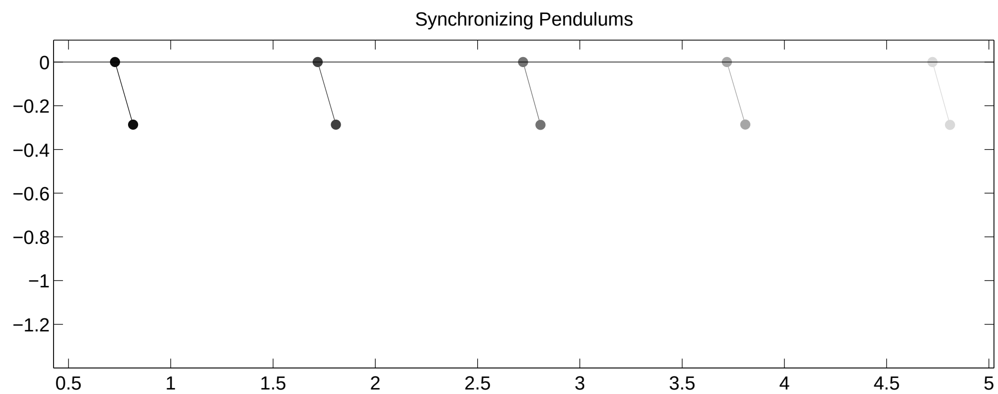
The dynamics of a single cart-pendulum system (referred to as an agent) can be derived using Lagrange's Equations (e.g. Spong (1989)). We can define the parameters and forces acting on the system through Figure 13(a). The only parameterized forces acting on the system, F_v and F_z, applied in the v and z directions, will cause oscillations of the θ and ψ angles. These forces will be the control inputs, u_v and u_z, to the system. No damping force is considered in this model as pendula can be approximated as zero damping systems. However, these dynamics cannot be linearized about the $\theta = 0$ hanging equilibrium point because this point represents a singularity in the Jacobian of the system. As a result, Yang et al. (2000) propose an approximation of the dynamics given that the angle from vertical, θ stays under 10 degrees. When this is the case, the pendulum can be projected onto the v-w and z-w planes such that new angles, α and β, can be defined. This separates the system such that β changes purely as a result of u_z and α as a result of u_v. For small angles, this approximation assumes the projected pendulum length is l.



(a) Initial Condition.



(b) Distributed case After 25s.



(c) Centralized case After 25s.

Figure 9. Animation of 5 pendula showing initial conditions and final synchronization for both distributed and centralized cases.

Figure 13(b) shows the resulting system and the resulting equations of motion are

$$\begin{aligned}
\ddot{v} =& -\frac{ml\ddot{\alpha}}{M+m}cos(\alpha) + \frac{ml\dot{\alpha}^2}{M+m}sin(\alpha) + \frac{u_v}{M+m} \\
\ddot{z} =& -\frac{ml\ddot{\beta}}{M+m}cos(\beta) + \frac{ml\dot{\beta}^2}{M+m}sin(\beta) + \frac{u_z}{M+m} \\
\ddot{\alpha} =& -\frac{\ddot{v}}{l}cos(\alpha) - \frac{g}{l}sin(\alpha) \\
\ddot{\beta} =& -\frac{\ddot{z}}{l}cos(\beta) - \frac{g}{l}sin(\beta).
\end{aligned}$$

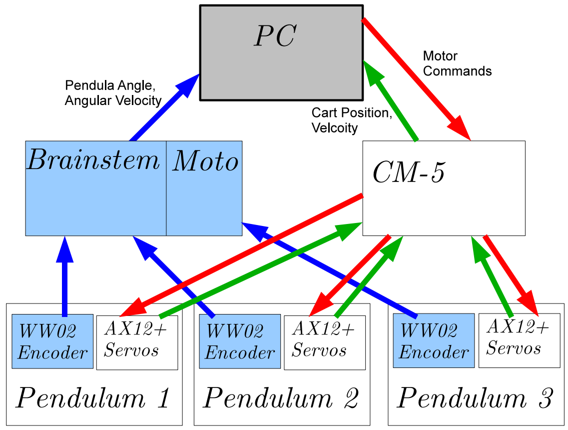
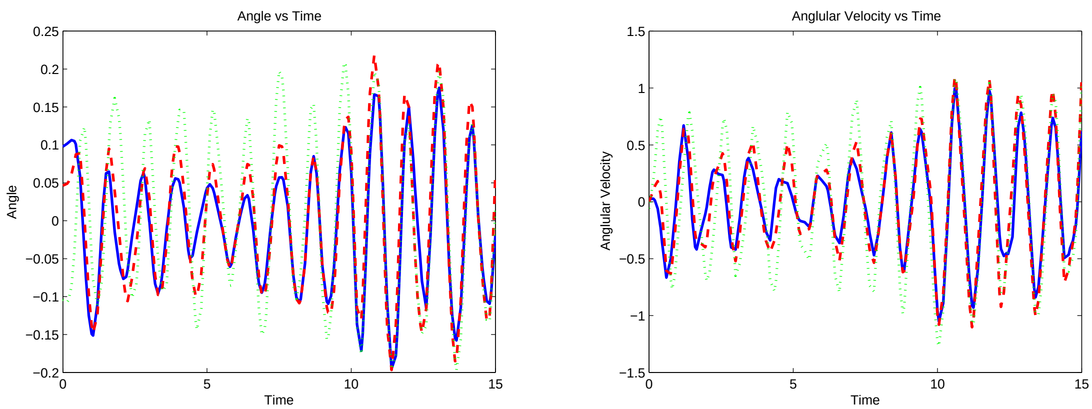


Figure 10. System architecture of 3 pendula platform.



(a) Pendulum Angle.

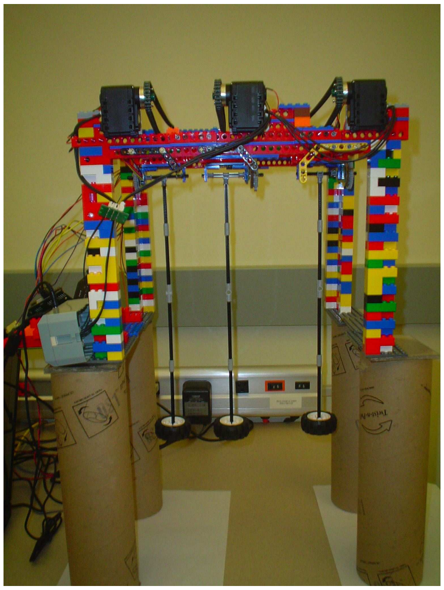
(b) Pendulum Angular Velocity.

Figure 11. Demonstration results showing synchronization of experimental pendula.

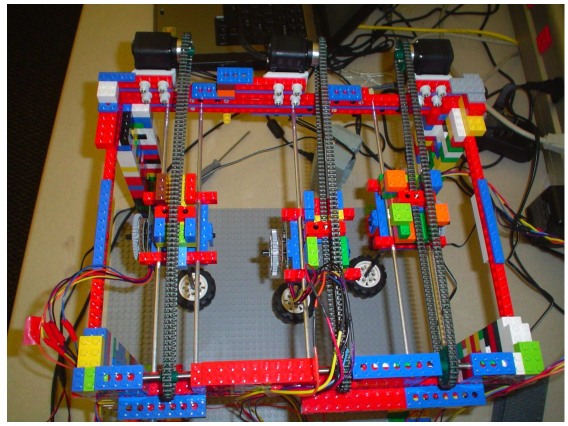
The resulting single pendulum system linearized about the $\alpha = 0$, $\dot{\alpha} = 0$, $\beta = 0$, $\dot{\beta} = 0$, $\dot{v} = 0$, $\dot{z} = 0$ equilibrum point is $\dot{x}_{ii}(t) = A_i x_{ii}(t) + B_i u_i(t)$, where

$$x_{ii} = \begin{bmatrix} v_i \\ \dot{v}_i \\ z_i \\ \dot{z}_i \\ \alpha_i \\ \dot{\alpha}_i \\ \beta_i \\ \dot{\beta}_i \end{bmatrix}, B_i = \begin{bmatrix} 0 & 0 \\ \frac{1}{M} & 0 \\ 0 & 0 \\ 0 & \frac{1}{M} \\ 0 & 0 \\ \frac{-1}{Ml} & 0 \\ 0 & 0 \\ 0 & \frac{-1}{Ml} \end{bmatrix}$$

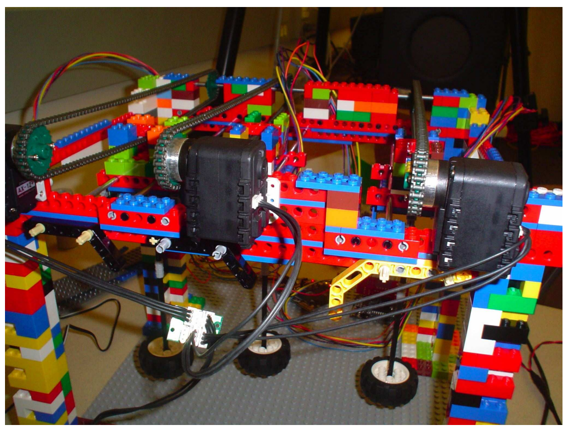
$$A_i = \begin{bmatrix} 0\,1\,0\,0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,1 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 1 & 0 & 0 \\ 0\,0\,0\,0 & \frac{-(M+m)g}{Ml} & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 0 & 0 & 1 \\ 0\,0\,0\,0 & 0 & 0 & \frac{-(M+m)g}{Ml} & 0 \end{bmatrix}.$$



(a) Demonstration testbed.



(b) Carts.



(c) Servo Motors.

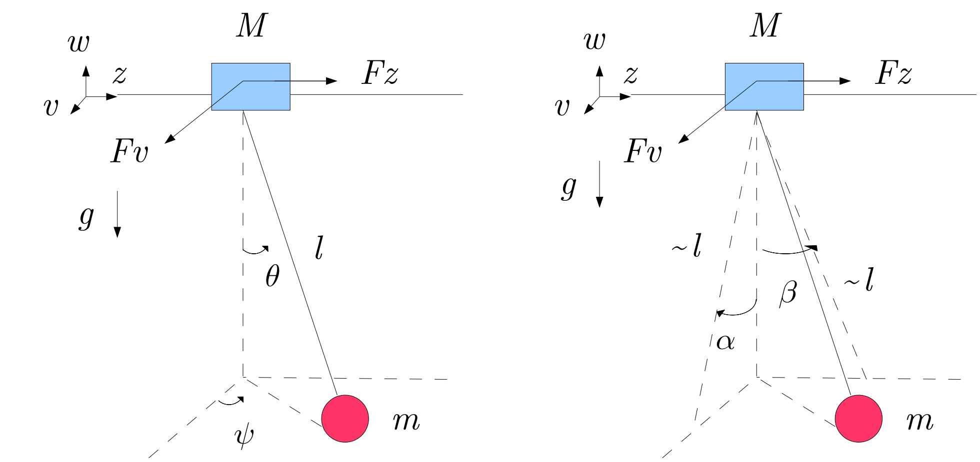
Figure 12. Demonstration Platform Images.

Note that this pair, (A_i, B_i), is completely controllable.

For a N planar pendulum system, the system can be written as $\dot{x}(t) = Ax(t) + Bu(t)$, where

$$
\begin{aligned}
x(t) &= \left[x_{11}^T(t), \ldots, x_{NN}^T(t)\right]^T, \qquad (29)\\
u(t) &= \left[u_1(t) \ldots u_N(t)\right]^T,\\
A &= \begin{bmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_N \end{bmatrix}, B = \begin{bmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_N \end{bmatrix}.
\end{aligned}
$$

Recall that in this paper, $A_i = A_j$ and $B_i = B_j$ for $i, j = 1, \ldots, N$, since the pendula are assumed to be homogeneous. For this example, we will be using the example graph, G, as our a representation of the distributed cart-pendula system. Given this system, a control law is sought to drive these pendula in such a way that they achieve identical angles, angular velocities, and cart velocities while maintaining a set distance, δ, between the carts. This requirement will



(a) Pendulum Diagram with hanging singularity. (b) Modified Pendulum Diagram with small angle approximation.

Figure 13. Pendulum Diagrams

be the pairwise constraints enforced between adjacent agents i and j such that (3) is enforced, for

$$P = \left[I_{8\times 8}\right] \tag{30}$$

$$b = \left[\delta_1 \; 0 \; \delta_2 \; 0 \; 0 \; 0 \; 0 \; 0\right]^T \tag{31}$$

with distances $\delta_1 \in \mathbb{R}^1$ in the P_x direction and $\delta_2 \in \mathbb{R}^1$ in the P_z direction such that $\delta = \sqrt{\delta_1^2 + \delta_2^2}$. $I_{8\times 8}$ is the identity matrix. It should be noted that δ can be different for each pairwise constraint on the graph.The assumptions made about this system are identical to the assumptions made in the planar pendula case in the last section.

5.2 *System Control: Consensus Algorithm + Optimal Control*

Now, that the system dynamics and pairwise constraints are defined for a given network graph, we define the consensus problem to be the same as the planar case. The only difference is that we are now dealing with a larger class of network topologies, trees graphs instead of just line graphs. Let d_i indicate the number of adjacent agents to agent i for all $i \in V_p$. Each pendulum can calculate state opinions for its neighbors based on its state, $x_{ii}(t) \in \mathbb{R}^8$, and the state of its d_i adjacent agents, $x_{jj}(t) \in \mathbb{R}^8$ for $j \in \mathbb{N}_i$. This local state opinion vector is defined as $X_{d_i}(t) = [y_1, \ldots, x_i, \ldots, y_{d_i}]^T \in \mathbb{R}^{8\times N}$, where

$$y_k = \left\{ x_{jj}(t) \text{ where } j \text{ is the } k\text{th agent in } \mathbb{N}_i \right. \tag{32}$$

for $k = 1, \ldots, d_i$ and $j = 1, \ldots, N$. For agent i, A_{d_i} and B_{d_i} are the corresponding pendula state space models for d_i adjacent agents. $W_{d_i}(t)$ is the Grammian for the pair (A_{d_i}, B_{d_i}) from time t to T.

Here, each pendulum can initially solve for a optimal terminal state based on the local state opinion vector that satisfies its local constraint at $t = 0$. Let τ_i be the matrix T_i with all columns with every element being zero removed. From Section 5, the optimal terminal state, used to initialize the state opinions for adjacent agents, is given by the unique solution,

$$x_{iT} = Q_i^{-1}(-R_i^T + C_i^T(C_iQ_i^{-1}C_i^T)^{-1}(p_i + C_iQ_i^{-1}R_i^T), \tag{33}$$

where

$$Q_i = 2e^{-A_{di}^T T} W_{di}^{-1} e^{-A_{di}T} \tag{34}$$

$$R_i = -2X_{d_i}^T(0)W_{di}^{-1}e^{-A_{di}T} \tag{35}$$

$$C_i = \tau^i \otimes P \tag{36}$$

$$p_i = f_i \otimes b. \tag{37}$$

The state opinion vector, x_i, for this problem will be defined using x_{iT} for adjacent agents state opinions and the state assignments in (16) for non-adjacent agent state opinions.

Then, $X_i(0)$, the initial conditions of the consensus equation, are defined from x_i and γ. After this, the consensus equation updates the state opinion vector, $X_i(t)$, for each pendulum based on the state opinion vectors of its neighbors. The state opinions for agent i can be extracted from the state opinion vector, $X_i(t)$, and used to calculate an optimal control law to drive agent i. After every update, the new consensus values of agent i are used in the optimal control presented in Chipalkatty et al. (2009). The control law, at time t is given by (26) for $i = 1, \dots, N$ and the control input for agent i is $u_i \in \mathbb{R}^2$. We now have a control that drives the entire network to a terminal state that satisfies the terminal constraint.

It should be noted that the pendula converge to a set distance δ apart and have equal angles and angular velocities. They also have equal cart velocities; however, these velocities are not guaranteed to be zero. It should also be noted that in order for this distributed control strategy to be effective, the consensus algorithm must converge to the agreement value before the specified final time in the optimal control law, T.

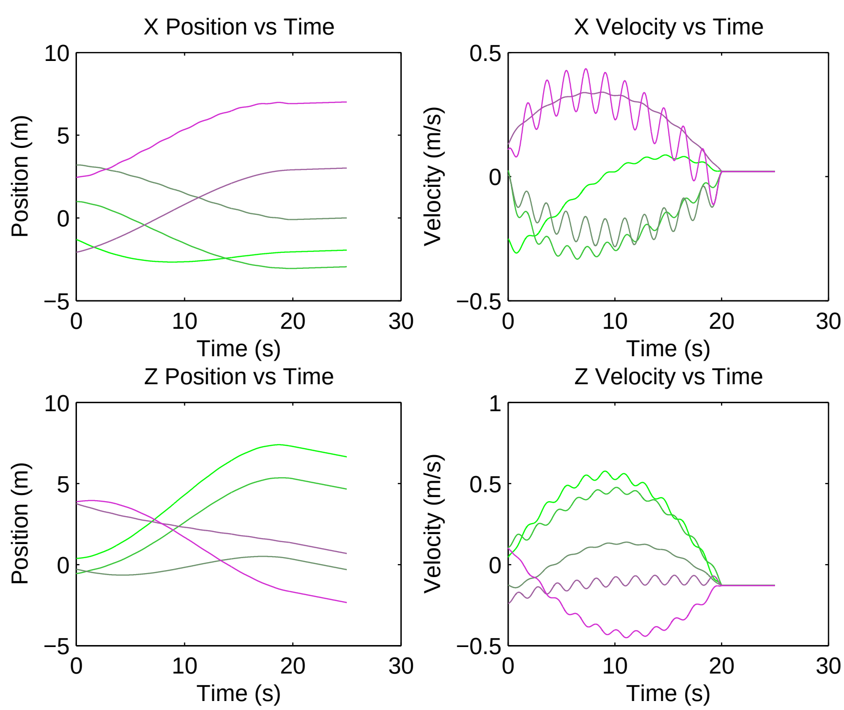


Figure 14. 5 Pendulum simulation results showing position formation and identical velocity. X and Z Positions plots show that the adjacent agents maintain inter-agent distance while the velocity plots show identical velocities after the 20 s mark.

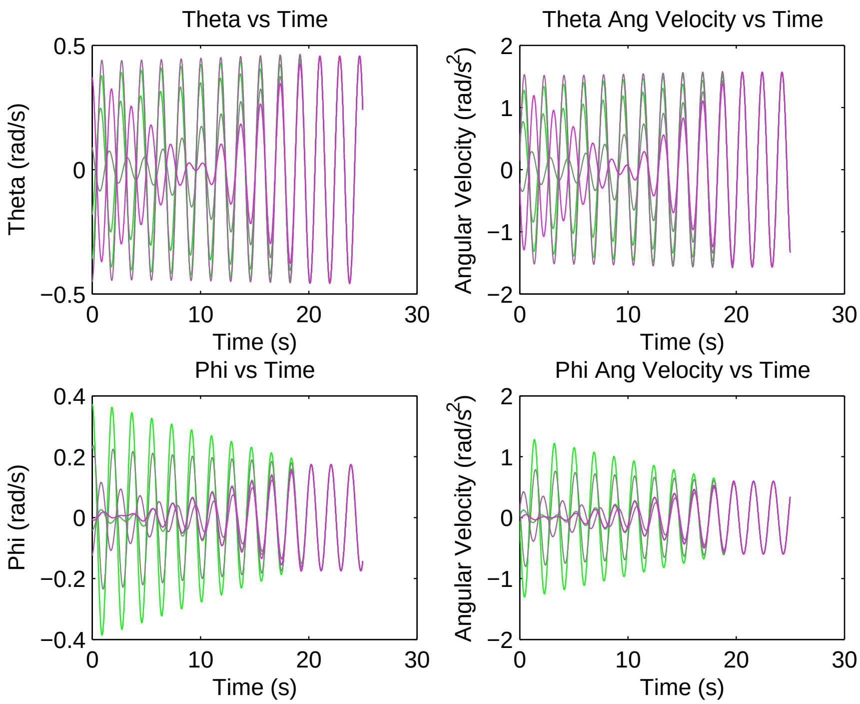


Figure 15. 5 Pendulum simulation results showing identical angles and angular velocity after the 20 s mark showing pendulua synchronization.

5.3 *Simulation Results*

The stated control laws are implemented in a MATLAB simulation of the presented pendulum dynamics. Simulations are run with the following parameters: $g = 9.8\ m/s$, $l = 0.30\ m$, $M = 1\ kg$, and $m = 0.2$ kg for the pendulum model. For each edge, a different δ was assigned: edge 1 ($\delta_1 = 1.0$ m, $\delta_2 = 2.0$ m), edge 2 ($\delta_1 = -5.0$ m, $\delta_2 = 6.0$ m), edge 3 ($\delta_1 = 3.0$ m, $\delta_2 = 1.0$ m), and edge 4 ($\delta_1 = -4.0$ m, $\delta_2 = 3.0$ m) It should be noted that in order for this distributed control strategy to be effective, the consensus algorithm must converge to the agreement value before the specified final time in the optimal control law, which in this case is 20 seconds.

In Figures 14 and 15, the results are shown for a five pendula scenario using the optimal control law and the consensus algorithm.

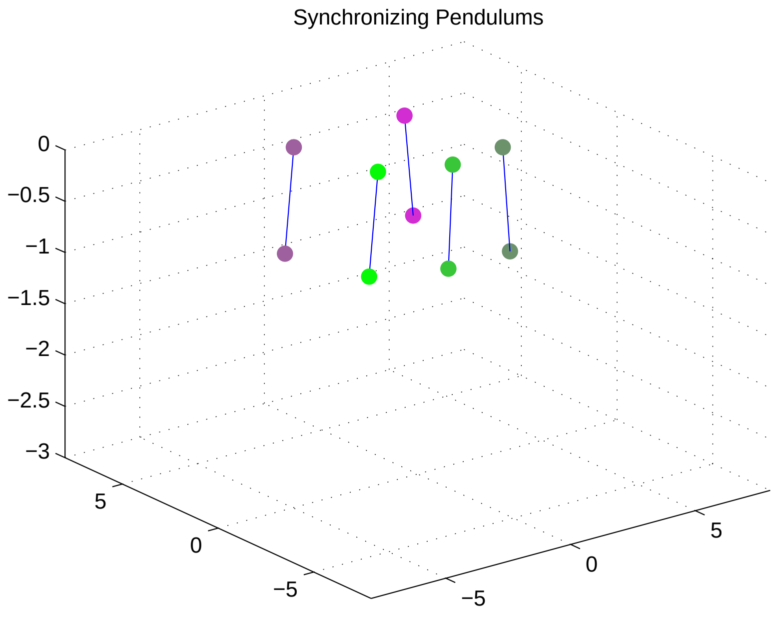
It can be seen that at 20 seconds, the distance between adjacent pendula is close to the respective δ's, as prescribed, while the velocities, angles, and angular velocities are identical for all the pendula. The animations of these scenarios are given in Figures 16(a) and 16(b).

6 Conclusions

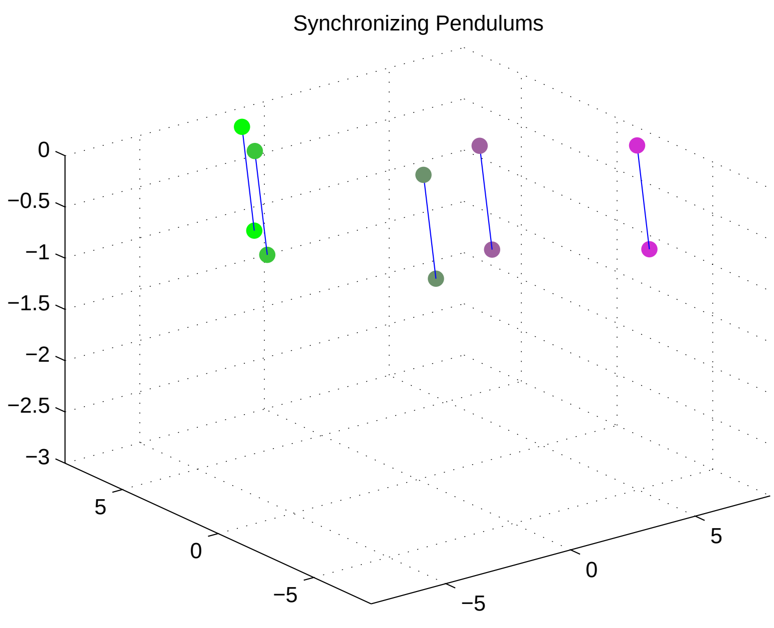
The main contribution of this paper is a method for reaching agreement among distributed agents (organized in a tree graph topology) such that a globally defined constraint is satisfied over the network given only local information. This method only requires a priori knowledge of the network topology and a static update law, that only needs to be initialized and updated using local information.

In other words, we just need to set-up the each agent's initial state opinion vector, which can be done using information communicated only from its neighbors, and then simply run the standard consensus equation (as given in Mesbahi and Egerstedt 2010) over the network. The result of the consensus equation is a network state that satisfies the global constraint.

This technique was applied to the control of a distributed network of linearized planar and spherical pendula. This algorithm, together with point-to-point transfer optimal control, was



(a) 5 Pendulum Simulation showing initial conditions.



(b) 5 Pendulum Simulation showing synchronization after 25s.

Figure 16. 5 Pendulum Simulation

able to drive these linear systems to a terminal manifold such that the manifold synchronizes the pendula oscillations and maintains a desired cart formation.

Future research directions include

Acknowledgments

This work was supported by the US National Science Foundation through Grant # 0820004. The authors would also like to thank Dr. Patrick Martin for the use of the software platform underpinning the experimental platform.

References

Bamieh, B., Paganini, F., and Dahleh, M. (2002), "Distributed control of spatially invariant systems," *Automatic Control, IEEE Transactions on*, 47, 1091–1107.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2005), "Gossip algorithms: Design, analysis, and applications," in *Proceedings IEEE Infocomm*, March, Vol. 3, Miami, pp. 1653–1664.

Chipalkatty, R., Egerstedt, M., and Azuma, S.I. (2009), "Multi-Pendulum Synchronization Using Constrained Agreement Protocols," in *RoboComm Conference*.

Dimarogonas, D., and Kyriakopoulos, K. (2007.), "Decentralized Navigation Functions for Multiple Robotic Agents with Limited Sensing Capabilities," *Journal of Intelligent and Robotic Systems*, 48, 411–433.

Jadbabaie, A., Lin, J., and Morse, A.S. (2003), "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, 48, 988–1001.

Jadbabaie, A., Motee, N., and Barahona, M. (2004), "On the stability of the Kuramoto model of coupled nonlinear oscillators," in *Proc. American Control Conference*, Vol. 5, pp. 4296–4301 vol.5.

Mesbahi, M., and Egerstedt, M., *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press (2010).

Moore, K.L., and Lucarelli, D. (2005), "Forced and constrained consensus among cooperating agents," in *Proc. IEEE Networking, Sensing and Control*, 19–22 March, pp. 449–454.

Motee, N., and Jadbabaie, A. (2008), "Optimal control of spatially distributed systems," *Automatic Control, IEEE Transactions on*, 53, 1616–1629.

Nedic, A., Ozdaglar, A., and Parrilo, P.A. (2008), "Constrained Consensus," Technical report 2779, LIDS.

Olfati-Saber, R., and Murray, R.M. (2003), "Agreement problems in networks with directed graphs and switching toplogy," in *Proc. 42nd IEEE Conf. Decision Control*, Dec, Vol. 4, Maui, HI, pp. 4126–4132.

Rantzer, A. (2009), "Dynamic dual decomposition for distributed control," in *American Control Conference, 2009. ACC'09.*, IEEE, pp. 884–888.

Ren, W., and Beard, R. (2004), "Consensus of information under dynamically changing interaction topologies," in *Proc. Am. Control Conf.*, Jun. 30-Jul. 2,, Vol. 6, pp. 4939–4944.

Rotkowitz, M., and Lall, S. (2006), "A characterization of convex problems in decentralized Control¡ sup¿*¡/sup¿," *Automatic Control, IEEE Transactions on*, 51, 274–286.

Spong, M.W., *Robot Dynamics and Control*, New York, NY, USA: John Wiley & Sons, Inc. (1989).

Tanner, H., Jadbabaie, A., and Pappas, G. (2000), "Coordination of multiple autonomous vehicles," in *Proceedings of the 11th IEEE Mediterranean Conference on Control and Automation*, June, Rhodes, Greece.

Xiao, L., Boyd, S., and Lall., S. (2005), "A scheme for robust distributed sensor fusion based on average consensus," in *International Conference on Information Processing in Sensor Networks*, Los Angeles, pp. 63–70.

Yang, R., Kuen, Y.Y., and Li, Z. (2000), "Stabilization of a 2-DOF spherical pendulum on X-Y table," in *Proc. IEEE International Conference on Control Applications*, 25–27 Sept., pp. 724–729.