Switching Rules for Decentralized Control with Simple Control Laws

Brian Shucker, Todd Murphey, and John K. Bennett
College of Engineering, 430 UCB
University of Colorado
Boulder, Colorado 80309
Email: {shucker, jkb, murphey}@colorado.edu

***Abstract*— We introduce a novel method for enforcing stability on a decentralized control system. In contrast to previous work, our approach allows for the use of a wide variety of simple control laws, while still providing for a formal proof of stability. Our motivating example uses a simple geometric switching function coupled with PD control that has an intuitive interpretation as a virtual spring mesh. Building on this example, we show a general proof technique that applies to a large class of decentralized control systems. Furthermore, we describe additional cases that illustrate how our technique can be applied to useful systems that are straightforward to implement.**

I. Introduction

We have developed a novel method for enforcing stability on a decentralized control system. Many decentralized control algorithms are modeled after phenomena observed in nature, such as the flocking behavior of birds or the schooling behavior of fish [1], [2]. Others are based on simulated physical systems, such as cellular automata in crystals [3] or biological cells. Common to these approaches are simple local control laws implemented on each agent, and designed in such a way that desirable global behaviors emerge. The control laws are typically based on interactions between a given agent, the environment, and a set of nearby agents determined by a proximity graph of some kind.

Proofs of stability have been produced for such systems (e.g., [4], [5]), but typically these proofs impose constraints on the dynamics of the system and the proximity graph. For example, the results in [4] apply only to a specific potential function on the unit-disk graph, and the results in [5] are for another particular potential function on a Voronoi graph. The difficulty with this is that the stability results do not leave much room for task specification–tasks must be framed in terms of what can be achieved in a stable manner and are therefore often limited to stable area coverage or "flocking" through a series of obstacles. Moreover, the task specification will likely change over time, thus introducing discrete changes into the equations of motion. Lastly, heuristics of various sorts are often helpful for various tasks, such as collision avoidance and other safety-critical elements of the task specification. The key point is that the control should not dictate the task specification, or it should at least minimize its effect. To this end, we explore a more general method of proving stability with an eye towards ease of implementation, genericity of proximity graphs to which it is applicable, and provable stability.

Traditional approaches to stability of hybrid switching systems typically require that one find a common Lyapunov function for all possible hybrid states of the system [6]. This is often an intractable problem for systems with large numbers of hybrid states. It is no coincidence that coordinated control systems are such systems, having as many as $(n-1)!$ states for n agents. Dwell-time analysis, such as that found in [7], seeks to provide stability for a more general class of systems by imposing restrictions on the (global) switching rate. Our work builds upon these initial results. However, it is not clear how to apply this type of analysis to a distributed system, where each agent has access to only local information. This is part of what is considered here, and we show how to use a consensus algorithm as part of the hysteresis-generating function to decentralize the approach.

Our work seeks to take advantage of the intuition behind dwell-time analysis and produce a general technique for proving stability using only local information. We are motivated by a desire to construct a distributed system where we choose the proximity graph based on desired topological properties, and then adapt the resulting switching function to gain provable stability.

Throughout this paper, our motivating example will be the switching function that produces a Gabriel graph, which we describe in more detail in [8] and [9]. This Gabriel graph switching function is described briefly in Section III. Section IV details the generalized result that applies to a large class of systems, including our motivating example. Sections VI and VII present additional examples of common situations where our technique may be put to use. We end with a discussion of some of the strengths and weaknesses of our approach as well as future work.

II. Related Work

There is a significant body of previous work dealing with coordination of small teams of robots, e.g.[10], [11], [12], [13], [14], [15]. More recently, there has been research into behavior-based and virtual-physics based control of large teams of robots[16], [17], [18], [19], [20], [21]. The work most closely related to our own is summarized below.

A. Behavior-based Control

Fully distributed control based upon simple local behaviors has been used in several contexts. Much of this research is based on the intuition gained from observing behaviors such as flocking in animals. In flocking situations, animals seem to draw most of their behavioral cues from the nearby flockmates. Using this observation as a basis, Brooks[16] has investigated behavior-based control extensively; Werger[17] later described the design principles of such systems. Balch and Hybinette[3] suggested the use of "attachment sites" that mimic the geometry of crystals; this is used to create formations with large numbers of robots. A variety of projects have made use of "swarm robotics," e.g., [22] and [23], to carry out simple tasks such as light tracking. Gage[18] investigated the use of robot swarms to provide blanket, barrier, or sweep coverage of an area. Several researches have used models based on the interactions of ants within a colony[24], [23], [25]. These approaches generally seek to define simple local behaviors that lead to large-scale properties that are beneficial in a particular application.

B. Virtual Physics

Distributed control based on virtual physics (also called "artificial physics" or "physicomimetics") has also been investigated, although not in the manner described here. Howard, Mataric and Sukhatme[19] model robots as like electric charges in order to cause uniform deployment into an unknown enclosed area. Spears and Gordon[20], [21], [26] use a more sophisticated model analogous to the gravitational force, but make the force repulsive at close range. Both of these models use switching functions based on a threshold distance. McLurkin[27] used a partially-connected interaction graph with a physics model similar to that of compressed springs to produce uniform deployment within a limited indoor environment. These works provide useful heuristic algorithms, but unlike our work, they do not attempt to show any provable properties of the resulting formations.

C. Switched Systems

Jadbabaie and colleagues used algebraic graph theory to show stability for switched networks using nearby-neighbor rules[28], [29], [2]. Hespanha and Morse used dwell-time analysis to show stability in linear systems with arbitrary switching that is slow on the average[30], [7], [31]. Bullo and colleagues showed stability in a switched system using Voronoi neighbors[5]. Our work expands this prior work by allowing for more flexibility and a clear method of implementation in a distributed system.

III. Motivating Example: Virtual Physics Spring Mesh

In previous work [8], [9], we analyzed a distributed control system that was essentially a virtual physics model that looked like a spring mesh system. In this example, each agent is treated as a particle in a simulated system, with virtual springs acting between specific pairs of agents. The appeal of this control law is partially its conceptual simplicity and ease of implementation.

For a fixed set of springs, the control law for each agent is

$$\ddot{\mathbf{x}} = \mathbf{u} \tag{1}$$

$$\mathbf{u} = \big[\sum_{i \in C} k_s(l_i - l_0)\hat{\mathbf{v}}_{\mathbf{i}}\big] - k_d\dot{\mathbf{x}} \tag{2}$$

where $\mathbf{x}$ represents the Cartesian coordinates describing the agent's position, $\ddot{\mathbf{x}}$ is the agent's acceleration, $\dot{\mathbf{x}}$ is the agent's velocity, C is the set of springs connected to this agent, l_i is the length of the i^{th} spring, and $\hat{\mathbf{v}}_{\mathbf{i}}$ is the unit vector from this agent to the agent on the other end of the i^{th} spring. Control constants are the natural spring length (l_0), the spring stiffness (k_s), and the damping coefficient (k_d). Lastly, we require that the system be symmetric–if an agent a has a spring connected to agent b, then agent b must have a spring connected to agent a.

It is straightforward to show that such a system is stable in the absence of switching; that is, when springs are neither created nor destroyed (see [8]). However, it is useful to allow the creation and destruction of springs. In particular, when any sort of proximity graph is changing dynamically as the time evolution proceeds, springs will be created and destroyed.

Let R be the set of agents. Let the *sensor graph* G_S be a graph where R is the vertex set, and there is an edge between two vertices r_1 and $r_2 \in R$ iff agents r_1 and r_2 can both sense each other. Let the *control graph* G_C be a graph where R is the vertex set, and there is an edge between two vertices r_1 and $r_2 \in R$ iff agents r_1 and r_2 are interacting for control purposes. To simplify notation, we will understand S to be the edge set of G_S and C to be the edge set of G_C. C (and therefore G_C) will be defined by a time-varying switching function σ, which we will describe in terms of a graph construction algorithm. Note that C is necessarily a subset of S.

In our prior work, we introduced a switching function that creates a Gabriel graph G_C [32], [33], [34] that dictates which data is incorporated into the control laws. In particular, it dictates C. With this switching function, there is a spring between agents A and B if and only if for all other agents Z, the interior angle $\angle AZB$ is acute. Equivalently, there is a spring between agents A and B if there are no other agents within the circle with diameter $\overline{AB}$. This switching function uniquely determines C based on the agents' positions.

The Gabriel graph switching function provides many advantages; chief among these is provable connectivity of the graph [33]. The Gabriel graph is also well-suited to providing uniform coverage of an area, as it creates a mesh of acute triangles. It is a planar graph [33], so it does not suffer from high edge density when the agents are close together. However, it depends on springs being created with nonzero potential energy, which complicates any proof of stability, as energy may be added to the system as the topology changes.

In order to prove stability in the presence of time-varying topology, we modify the switching algorithm in a manner inspired by dwell-time analysis. It has been shown in several cases that if all members of a given class of linear systems are stable, then arbitrary switching among those systems results in a stable hybrid system, provided that the switching rate is "slow-on-the-average" [7]. Essentially, the proof shows that the rate of decrease of the Lyapunov function due to the dissipation is greater than the rate of increase of the Lyapunov function due to switching, as long as the average *dwell time* between switches is sufficiently long.

In our approach, instead of computing a limit on the switching frequency explicitly, we use a notion of a global "energy reserve" to create the same limiting effect on the switching rate. (The idea behind this name is that if a switch will increase the value of the Lyapunov function, there must be enough energy in reserve to compensate.) We find this approach intuitive and more straightforward to implement in our distributed system, in which switching events are detected locally. Although a global quantity such as this should make one nervous, we will see that a local estimate of this quantity based on a zero sum consensus algorithm is sufficient for stability purposes.

Consider a set of agents $r_i \in R$. Let the time-varying signal $\sigma : t \longrightarrow G$ be the switching function for a Gabriel graph G (i.e., σ determines the time evolution of G). Note that σ is constant except for discrete changes at times $t_1 ... t_n$. For any time interval $\tau_j = [t_j ... t_{j+1}]$, let $\mathbf{V}_{\sigma(\tau_j)}$ be a global potential function. It is shown in [8] that a function exists with the following properties:

1) $\mathbf{V}_{\sigma(\tau_j)}$ is positive-definite.
2) $\dot{\mathbf{V}}_{\sigma(\tau_j)}$ is negative semi-definite.
3) $\ddot{\mathbf{V}}_{\sigma(\tau_j)}$ is bounded.

These conditions imply (via Barbalat's lemma) that the system is stable during the intervals between switches. We define the overall potential function $\mathbf{V}_{\sigma(t)}$ to be equal to $\mathbf{V}_{\sigma(\tau_j)}$ on the interval $[t_j ... t_{j+1}]$, for all j. We will generalize this in Section IV.

Since it is possible to evaluate the potential associated with every spring at any time, each agent may maintain an estimate of the current potential of all springs connected to that agent. We will call this value $\mathbf{U}_i$.

$$\mathbf{U}_i = \sum_{h \in C_i} \frac{1}{2} k_s (l_{i,h} - l_0)^2 \tag{3}$$

where C_i is the set of springs connected to agent i, and $l_{i,h}$ represents the length of the spring from agent i to agent h. Whenever a switch occurs, the value of $\mathbf{U}_i$ may instantaneously change according to the potential created or destroyed by springs coming into and out of existence. Define the quantity s_i such that:

$$s_i(t) = \frac{1}{2} \left(\lim_{t \to \tilde{t}^+} \mathbf{U}_i - \lim_{t \to \tilde{t}^-} \mathbf{U}_i \right) \tag{4}$$

This quantity captures the instantaneous change in potential due to the spring switching. The factor of 1/2 is present because each spring connects to two agents, and thus will be counted twice. It is thus easy to show that the following equality holds:

$$\sum_{i \in R} s_i = \lim_{t \to \tilde{t}^+} (\mathbf{V}_{\sigma(t)}) - \lim_{t \to \tilde{t}^-} (\mathbf{V}_{\sigma(t)}) \tag{5}$$

Additionally, let

$$d_i = -k_d \dot{\mathbf{x}}_\mathbf{i}^\mathbf{T} \dot{\mathbf{x}}_\mathbf{i} \tag{6}$$

where $\mathbf{x_i}$ represents the position of agent i. The quantity d_i represents the rate of energy dissipated by damping at agent i. It is a direct consequence of the static stability proof in [8] that on any interval between switches, the following holds:

$$\sum_{i \in R} d_i = \dot{\mathbf{V}}_{\sigma(\tau_j)}. \tag{7}$$

This is clear if we recall the fact that the virtual physics is based on a spring mesh system, where all the energy dissipation is due to damping, and the total energy damped is the sum of the energy damped at each node of the mesh.

At this point, each agent can quantify its own contribution to the amount of energy that is being damped out of the system, as well as the amount that is being created or destroyed by switching. Intuitively, we would like the former to be of greater magnitude than the latter when averaged over all agents for some length of time.

This can be accomplished by maintaining a *local energy reserve* E_i at each agent (the local reserve will be related to a consensus-based global reserve in Section V). E_i is initialized to an arbitrary nonnegative value. As energy is damped out of the system, a fraction of that energy is added to the reserve. When a switch occurs, the energy created by the switch is removed from the reserve. As long as the energy reserve is not allowed to drop indefinitely, the system will be stable. This inspired us to create the *modified Gabriel graph switching function* $\sigma'(t)$, which is identical to $\sigma(t)$, except that a agent i may not create a spring if that operation would cause $E_i < 0$.

Notice that preventing the creation of a spring requires the cooperation of two agents (one on each end), since the properties of $\mathbf{V}_{\sigma(\tau_j)}$ given above depend on symmetry in the springs (that is G_C must be an undirected graph). Thus, spring creation is prohibited when either agent has $E_i < 0$.

A stability proof specific to a spring mesh with the modified Gabriel graph switching function is given in [8]. However, the underlying concept does not rely on that particular switching function, or on the spring mesh dynamics. The following section generalizes the proof in [8] to a broad class of systems, of which the Gabriel graph is a member.

IV. General Result

Consider a set of agents R and a time-varying switching signal $\sigma(t)$ that is constant except for discrete changes at times $t_1 .... t_n$. Assume that the state for each agent i is $\mathbf{x} \in M$, the governing equations are $\dot{\mathbf{x}} = f(\mathbf{x})$, and that the switching function changes f over time, $\sigma : (\mathbf{x}, t) \longrightarrow f$. We assume the following properties:

A1 For each time interval $[t_j ... t_{j+1}]$ (we will call this interval τ_j), there exists a global potential function $\mathbf{V}_{\sigma(\tau_j)}$ such that $\mathbf{V}_{\sigma(\tau_j)}$ is positive-definite, $\dot{\mathbf{V}}_{\sigma(\tau_j)}$ is negative semi-definite, and $\ddot{\mathbf{V}}_{\sigma(\tau_j)}$ is bounded. We define the overall potential function $\mathbf{V}_{\sigma(t)}$ to be equal to $\mathbf{V}_{\sigma(\tau_j)}$ on the interval $[t_j ... t_{j+1}]$, for all j.

A2 At every time t, each agent i can determine a quantity d_i such that $\dot{d}_i$ is bounded, $\sum_{i \in R} d_i \geq \dot{\mathbf{V}}_{\sigma(t)}$ and $d_i \leq$

0. Note that $\dot{\mathbf{V}}_{\sigma(t)}$ is negative semi-definite, so d_i is bounded above by zero and below by $\dot{\mathbf{V}}_{\sigma(t)}$.

A3 At every time t, let there be a quantity s_i for each agent such that $\sum_{i\in R} s_i = \lim_{\tilde{t}\to t^+}(\mathbf{V}_{\sigma(\tilde{t})}) - \lim_{\tilde{t}\to t^-}(\mathbf{V}_{\sigma(\tilde{t})})$. Each agent can determine an estimate $\hat{s}_i$ such that $\sum_{i\in R}\hat{s}_i \geq \sum_{i\in R} s_i$.

A4 A switch at time t_j for which $\hat{s}_i > 0$ for any $i \in R$ may be prohibited at will, causing $\sigma(\tau_j) = \sigma(\tau_{j-1})$.

Property **A1** implies that the system is stable in the absence of switching. This is typically simple to verify using standard Lyapunov function techniques.

Property **A2** involves the agents' local estimates of the amount of energy that is being damped out of the system. If Property **A2** is satisfied, then the sum of the local estimates does not collectively over-estimate the amount of damping that occurs.

Property **A3** involves the agents' local estimates of the potential created by switching. If Property **A3** is satisfied, then the sum of the local estimates does not collectively under-estimate the actual potential created by a switch.

Property **A4** captures the ability of the agents to delay or prevent a switch when necessary to prevent destabilization.

If each of these properties is satisfied, then our method is applicable and the system may be stabilized with a simple modification to the switching function.

Associate with each agent i a value E_i which is defined as the solution to a differential equation. E_i has an arbitrarily chosen nonnegative initial value and evolves according to the following:

$$\dot{E}_i(t) = -k_e d_i(t) \text{ if } s_i(t) = 0 \tag{8}$$

$$E_i(t) = \lim_{\tilde{t}\to t^-} E_i(\tilde{t}) - s_i(t) \text{ otherwise} \tag{9}$$

where k_e is a global constant, $0 < k_e < 1$. E_i represents the *local energy reserve*. Notice that E_i is initialized to a nonnegative value and then evolves according to Equation 8 as long as s_i is zero (that is, on intervals with no switches). Whenever $s_i \neq 0$ (there is a switch), E_i is re-initialized to the value given in Equation 9.

Each agent maintains a local estimate $\hat{E}_i$, which is initially greater than zero and evolves according to the following:

$$\dot{\hat{E}}_i(t) = -k_e d_i(t) \text{ if } \hat{s}_i(t) = 0 \tag{10}$$

$$\hat{E}_i(t) = \lim_{\tilde{t}\to t^-} \hat{E}(\tilde{t}) - \hat{s}_i(t) \text{ otherwise} \tag{11}$$

Let the global values E and $\hat{E}$ be defined such that

$$E = \sum_{i\in R} E_i \tag{12}$$

$$\hat{E} = \sum_{i\in R} \hat{E}_i \tag{13}$$

We will call E the *global energy reserve*.

This brings us to the simple change necessary to stabilize the system. The *modified switching function* σ' is identical to σ, except for the added condition that any switch that would cause $\hat{E}_i < 0$ for any agent i is prohibited, as described in Property **A4**. Note that the value of $\hat{E}_i$ cannot decrease in the absence of switching, because $d_i \leq 0$ for all i (see Property **A2**). Also, this computation is decentralized; the agents only need access to the local values E_i, d_i, and s_i.

The immediate consequence of modifying σ in this way is that $\hat{E} \geq 0$, since it is the sum of all nonnegative terms. It follows from equations 12 and 13 and the definitions of s_i and $\hat{s}_i$ (see Property **A3**) that $E \geq \hat{E}$. Thus if $\hat{E} \geq 0$, then $E \geq 0$ as well.

Theorem 4.1: In any system satisfying assumptions **A1**-**A4** and using the modified switching function σ', all agents eventually reach a state of unchanging potential. That is, $|\mathbf{V}_{\sigma'(t)} - \alpha| \to 0$ for some $\alpha \in \mathbb{R}$ and, in particular, $\dot{\mathbf{V}}_{\sigma'(t)} \to 0$.

For purposes of notational simplicity, we will take $\mathbf{V}$ to denote $\mathbf{V}_{\sigma'(t)}$ for the remainder of this section unless otherwise specified.

Proof: Our approach invokes Barbalat's lemma, which states that if $f(t)$ is lower bounded, $\dot{f}(t)$ is negative semi-definite, and $\dot{f}(t)$ is uniformly continuous (or equivalently, $\ddot{f}(t)$ is finite), then $\dot{f}(t)$ approaches zero as t approaches infinity. We will apply Barbalat's lemma to a potential function $\mathbf{V}'$, thereby showing that $\dot{\mathbf{V}}'$ goes to zero, which implies that all agents reach a state of unchanging potential.

We will show stability of the system using the *modified potential function* $\mathbf{V}'$, defined as:

$$\mathbf{V}' = \mathbf{V} + E \tag{14}$$

Since $\mathbf{V}$ is positive-definite (by Property **A1**) and $E > 0$, it is clear that $\mathbf{V}' \geq 0$.

Differentiating, we see that on any interval on which there are no switches:

$$\dot{\mathbf{V}}' = \dot{\mathbf{V}} + \dot{E} \tag{15}$$

Substituting for $\dot{E}$:

$$\dot{\mathbf{V}}' = \dot{\mathbf{V}} + \sum_{i\in R} -k_e d_i \tag{16}$$

To handle switches, we must look back to the definition in Property **A3**:

$$\lim_{\tilde{t}\to t^+} \mathbf{V}(\tilde{t}) = \lim_{\tilde{t}\to t^-} \mathbf{V}(\tilde{t}) + \sum_{i\in R} s_i(t) \tag{17}$$

Thus, at any instant t when a switch occurs (that is, when any $s_i \neq 0$),

$$\lim_{\tilde{t}\to t^+} \mathbf{V}'(\tilde{t}) = \lim_{\tilde{t}\to t^-} \mathbf{V}(\tilde{t}) + \sum_{i\in R} s_i(t) + E(t) \tag{18}$$

Substituting for E from Equation 9,

$$\lim_{\tilde{t}\to t^+} \mathbf{V}'(\tilde{t}) = \lim_{\tilde{t}\to t^-} \mathbf{V}(t) + \sum_{i\in R} s_i(t) + \lim_{\tilde{t}\to t^-} E(\tilde{t}) - \sum_{i\in R} s_i(t) \tag{19}$$

which simplifies in the following way:

$$\lim_{\tilde{t}\to t^+} \mathbf{V}'(\tilde{t}) = \lim_{\tilde{t}\to t^-} \mathbf{V}(t) + \lim_{\tilde{t}\to t^-} E(\tilde{t}) \tag{20}$$

$$\lim_{\tilde{t}\to t^+} \mathbf{V}'(\tilde{t}) = \lim_{\tilde{t}\to t^-} \mathbf{V}'(\tilde{t}) \tag{21}$$

Thus, the discontinuity in $\mathbf{V}'$ has been removed, as the limits from both sides are the same. Since switches have no effect whatsoever on $\mathbf{V}'$, Equation 16 holds true at all times.

Since $\dot{\mathbf{V}}$ is negative definite (see Property **A1**), $0 < k_e < 1$, and $\dot{\mathbf{V}} < \sum_{i \in R} k_e d_i < 0$ (see Property **A2**), it must be the case that $\dot{\mathbf{V}}'$ is negative semi-definite.

Because $\ddot{\mathbf{V}}$ is bounded (Property **A1**) and $\dot{d}_i$ is bounded for all i (Property **A2**), we also know $\ddot{\mathbf{V}}'$ is bounded.

We now have sufficient information to satisfy Barbalat's lemma. We know $\mathbf{V}'$ is lower bounded by zero, $\dot{\mathbf{V}}'$ is negative semi-definite, and $\ddot{\mathbf{V}}'$ is bounded, so Barbalat's lemma implies that $\dot{\mathbf{V}}' \to 0$ as $t \to \infty$. It follows directly that $\dot{\mathbf{V}}_{\sigma'(t)} \to 0$ as $t \to \infty$.

■

Note that in the proof of Theorem 4.1 we are effectively changing both where the switch in σ is allowed to occur and potentially which switches are allowed to occur.

V. Energy Reserve Consensus

Although the decision to prohibit a switch is made by each agent based on its local energy reserve, it is desirable to allow switches to occur whenever the *global* energy reserve is sufficiently large. That is, we do not want to prevent a switch due to low energy reserves in one part of the system, when there are sufficient energy reserves unused somewhere else. Thus, we need some mechanism for sharing information about the energy reserve levels between agents.

We will take advantage of the *average-consensus* algorithm described by Olfati-Saber and Murray [35]. This algorithm allows a distributed set of agents to reach a consensus on a common global value, while sharing information only with their local neighbors. If an agent i has a set of neighbors S_i which it can sense,

$$\bar{u}_i = \sum_{l \in S_i} (E_l - E_i) \tag{22}$$

We then replace Equations 8 and 10 with the following:

$$\dot{E}_i = -k_e d_i + \bar{u}_i \tag{23}$$

$$\dot{\hat{E}}_i = -k_e d_i + \bar{u}_i \tag{24}$$

Note that Equations 9 and 11 remain unchanged. We require that the neighbor relation is symmetric (if $a \in S_b$, then $b \in S_a$). This symmetry provides the following zero-sum property:

$$\sum_{i \in R} \bar{u}_i = 0 \tag{25}$$

Note that

$$\dot{E} = \sum_{i \in R} \dot{E}_i = \sum_{i \in R} -k_e d_i + \bar{u}_i = \sum_{i \in R} -k_e d_i$$

because of the zero-sum property. Hence, Equation 16 remains unchanged. Since Equation 9 is also unchanged, the result in Equations 19 through 21 also stands as before. The system does evolve differently, as the times when we must prohibit a switch have changed due to the differing *local* values of E, but it still meets all the conditions necessary for the proof in Section IV because the *global* behavior of E still has the required properties. However, as described in [35], all of the local energy reserves will now converge to a single value, provided the sensing graph is connected. If the sensing graph is not connected, then each partition will converge to its own value. This is not ideal from a performance standpoint, but does not affect the stability property of the system.

The consensus function given here is just one example of a valid consensus function. In fact, any consensus algorithm with the zero-sum property described in Equation 25 is acceptable. The consensus on E is independent of the normal control of the system, although a faster consensus will improve performance in terms of convergence rate. What we have shown is the following:

Corollary 5.1: In any system satisfying assumptions **A1**-**A4** where Eqs.(22)-(24) replace Eqs.8 and 10, all agents eventually reach a state of unchanging potential.

VI. Example: Nearest Neighbors

One common switching function is the nearest-neighbors function, in which agents interact with all neighbors within some threshold distance. While proofs of stability for specific systems using nearest-neighbor rules exist (e.g. [2]), these proofs typically do not generalize. Our technique applies to a broad class of systems using nearest neighbor rules. For example, consider the system with the following control law:

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i \tag{26}$$

$$\mathbf{u}_i = \big[\sum_{j \in N_i} \nabla P(\mathbf{x}_i, \mathbf{x}_j) \big] - k_d \dot{\mathbf{x}}_i \tag{27}$$

where N_i is the set of neighboring agents within some threshold distance of agent i and P is some continuous, conservative function representing the potential between agents. "Conservative" here is used in the sense of a conservative field–the integral over any two paths with the same endpoints is the same. ∇P is the gradient of P with respect to $\mathbf{x}$.

For each interval τ_j between switches, let the potential function be:

$$\mathbf{V}_{\sigma(\tau_j)} = \sum_{i \in R} \big[\sum_{j \in N_i} P(\mathbf{x}_i, \mathbf{x}_j) + \dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i \big] \tag{28}$$

Since P is conservative, it can be shown that:

$$\dot{\mathbf{V}}_{\sigma(\tau_j)} = \sum_{i \in R} -k_d \dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i \tag{29}$$

Thus, this definition of $\mathbf{V}_{\sigma(\tau_j)}$ satisfies Property **A1**.

The damping term makes it easy to satisfy Property **A2**; we simply let:

$$d_i = -k_d \dot{\mathbf{x}}_\mathbf{i}^\mathbf{T} \dot{\mathbf{x}}_\mathbf{i} \tag{30}$$

Since we can evaluate P at any point, satisfying Property **A3** is also straightforward. We define s_i such that:

$$s_i = \sum_{j \in N_i^+} P(\mathbf{x}_i, \mathbf{x}_j) - \sum_{j \in N_i^-} P(\mathbf{x}_i, \mathbf{x}_j) \tag{31}$$

where N_i^+ represents the limit of N_i from the right, and N_i^- represents the limit of N_i from the left.

Property **A4** is true because agents may agree not to interact with each other at will. Having met all the conditions, we apply our technique to construct a stable system with a modified switching function.

We have shown the following.

Corollary 6.1: With the nearest neighbor graph topology from (27) where σ' is substituted as described in Section IV, all agents eventually reach a state of unchanging potential.

VII. Example: Target Tracking

Consider a system in which there are potentials between the agents as well as between the agents and targets in the environment. For example, one might model agents as positive charges and targets as negative charges (similar to [19]), so that the agents normally disperse but are attracted to target areas. It may be the case that targets can appear, disappear, change position, and/or change characteristics in such a way as to inject large amounts of energy into the system. Our technique can be applied to prevent destabilization of the system due to target behavior.

For example, let R be a set of agents and T a set of targets, each of which may appear and disappear arbitrarily. Let the control law for agent i be the following:

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i \tag{32}$$

$$\mathbf{u}_i = \big[\sum_{j\in R} \nabla P_R(\mathbf{x}_i, \mathbf{x}_j)\big] + \big[\sum_{k\in T} \nabla P_T(\mathbf{x}_i, \mathbf{x}_k)\big] - k_d \dot{\mathbf{x}}_i \tag{33}$$

where P_R is the potential function acting between the agents, and P_T is the potential function acting between agents and targets.

If there are no restrictions on the appearance of targets, then targets may inject an arbitrary amount of energy into the system. This is not desirable, as the continued appearance of targets, or the appearing and disappearing of a few targets in an unfortunate pattern, could destabilize the system and/or cause collisions between the agents. Modifying the switching function according to our technique will remove this problem.

Recalling section IV, Property **A1** is met because the potential functions are conservative and the system is damped (a proof of this is omitted for brevity, but is fairly straightforward). We can meet Property **A2** with the following reasonable choice:

$$d_i = -k_d \dot{\mathbf{x}}_\mathbf{i}^\mathbf{T} \dot{\mathbf{x}}_\mathbf{i} \tag{34}$$

Property **A3** is met because we can evaluate the potential functions P_R and P_T at any point. Thus, we can assign $s_i = P_T(\mathbf{x}_i, \mathbf{x}_k)$ when target k appears, and $s_i = -P_T(\mathbf{x}_i, \mathbf{x}_k)$ when target k disappears.

Property **A4** is met because $s_i > 0$ only when a target appears, and agents may elect not to track a given target if necessary.

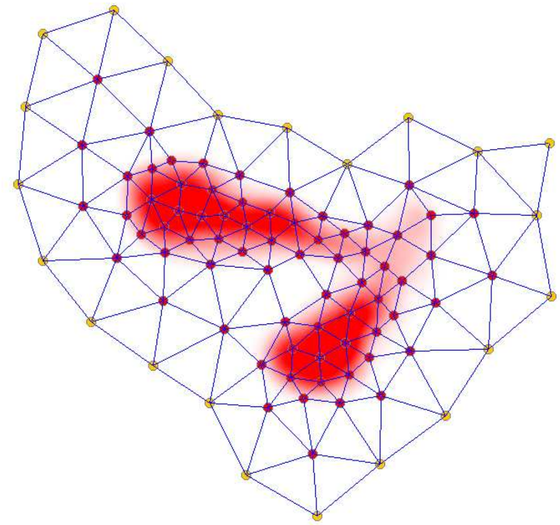

Fig. 1. Agents mapping a complex diffuse target

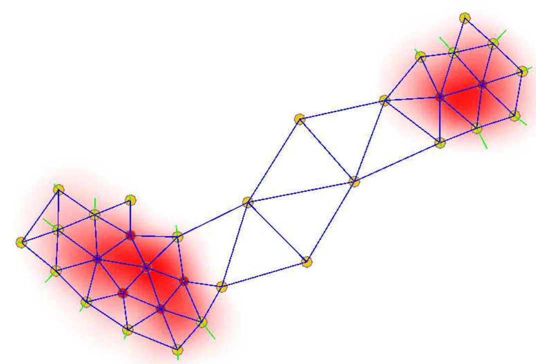

Fig. 2. Agents mapping multiple diffuse targets

Since each of the properties are satisfied, our technique may be applied. With our modified switching function, if the target pattern is ever such that the system would destabilize, then the agents will ignore the targets that would have caused destabilization to occur.

Further examples of target tracking are shown in Figures 1 and 2. In these cases, the targets are diffuse and represented by an intensity map. These simulations use the spring-mesh control law and Gabriel graph switching function from Section III, but with adaptive spring lengths so that the density of agents increases with the target intensity.

Figure 2 shows an interesting emergent property of the Gabriel graph algorithm. There are groups of agents tracking each target, and there are also some agents spread between the target areas to maintain connectivity. This is a useful formation, as it allows most of the available agents to be used for target tracking, but reserves some agents to maintain a communications path. The "division of labor" in this example is not explicit; it emerges as a result of the Gabriel graph switching function.

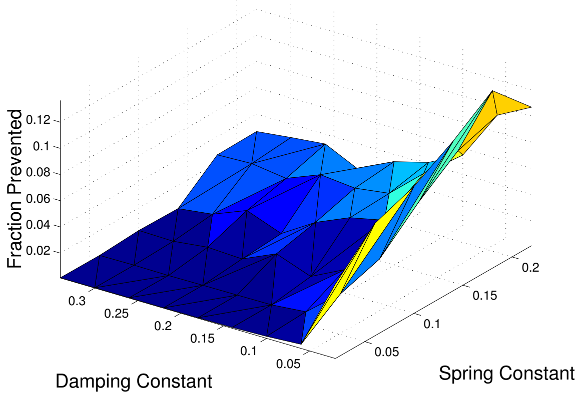

Fig. 3. Fraction of links prevented

Fig. 4. Target tracking experiment: the green robot (upper left) is the target. The remaining robots implement the control algorithm from Section III.

The combination of changing spring parameters and switching would normally be difficult to analyze. However, by applying our technique, it is possible to decouple the analysis of varying spring parameters from that of varying topology. As long as the system is stable in the absence of switching, we can ensure that the switching function does not introduce any instability.

VIII. Collision Avoidance

For some systems, using the modified switching function may have implications for collision avoidance. If its energy reserve is depleted, an agent may not allow a switch that is necessary in order to prevent a collision.

In such situations, it may be necessary to destroy an existing link or take other action in order to gain sufficient energy reserves to prevent the collision. The design of a switching function that guarantees the ability to do this is not obvious, and is an open area for future work.

IX. How Often Does σ' Play A Role?

Since modifying the switching function affects the behavior of the system, it is important to know just how often links are really prevented by insufficient energy reserves. One might expect that with conservative gains and damping, the energy recovered from damping will generally be great enough to cover the energy needs of switching. Only when operating with high gains and relatively little damping would one expect the energy reserve to truly come into play.

This is in fact what occurs. Figure 3 shows the fraction of links prevented by insufficient energy reserves for a test case with 32 robots using the modified Gabriel graph switching function. When the damping constant (k_d) is high and the spring constant (k_s) is low, no links are prevented. Only when the gain is relatively high and the damping constant is relatively low are there a large number of links prevented. This result is highly intuitive–when we "push the envelope," with higher gains, we are taking greater risks with stability. The modified switching function comes into play more and more as we push the system towards higher performance.

X. Hardware Experimentation

To verify the applicability of our results to real-world systems, we have implemented the system described in Section III in hardware. Our experimental platform is based on the Roomba robotic vacuum cleaner manufactured by iRobot. Each Roomba is outfitted with a controller board designed by the authors; this provides communication and implements the control algorithm on-board. A snapshot of one experiment is shown in Figure 4. In this experiment, six robots using the algorithm described in Section III track a single target, implemented by a seventh robot.

A detailed description of our hardware experiments can be found in [36].

XI. Conclusions

In this paper we have introduced an approach to cooperative control that focuses on monitoring the admissible changes in network graph topology according to a stability criterion. This method can be distributed across a network of agents by additionally using consensus algorithms like those found in [35]. This leads to a very flexible method of guaranteeing stability for arbitrary network graphs and explicitly avoids any instabilities due to the graph topology switching.

We did not consider collision avoidance or noise in this work, though the latter is largely addressed by the basic results of [7] on noise and external disturbances. Collision avoidance is the subject of continuing work. However, in simulation network links are always made in time to avoid collisions, so we believe our method will extend to this with guarantees on collision avoidance.

Finally, the basic results presented here have consequences for symbolic, linguistic, and grammatical control [37], [38]. Given a finite set of symbols, each corresponding to a stable process, Theorem 4.1 implies that so long as each symbol can "hold" its action in accordance with Eqs.(22)-(24), then any string of symbols will result in a stable system.

References

[1] C. Reynolds, "Flocks, birds, and schools: A distributed behavior model," *Computer Graphics*, vol. 21, pp. 25–34, 1987.

[2] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, “Stable flocking of mobile agents, Part II: Dynamic topology,” in *IEEE Conf. on Decision and Control*, 2003.

[3] T. Balch and M. Hybinette, “Behavior-based coordination of large-scale robot formations,” in *Proceedings of the Fourth International Conference on Multiagent Systems (ICMAS*, pp. 363–364, July 2000.

[4] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Transactions on Automatic Control*, 2003.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, 2004.

[6] D. Liberzon, *Switching in Systems and Control*. Birkhäuser, Boston, 2003.

[7] J. P. Hespanha and A. S. Morse, “Stability of switched systems with average dwell-time,” in *IEEE Conf. on Decision and Control*, 1999.

[8] B. Shucker, T. Murphey, and J. K. Bennett, “An approach to switching control beyond nearest neighbor rules,” in *American Control Conference*, 2006.

[9] B. Shucker, T. Murphey, and J. K. Bennett, “A method of cooperative control using occasional non-local interactions,” in *IEEE Conference on Robotics and Automation*, 2006.

[10] T. Balch and L. E. Parker, *Robot Teams: From Diversity to Polymorphism*. A K Peters Ltd, 2002.

[11] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926–939, 1998.

[12] T. Balch and R. C. Arkin, “Motor schema-based formation control for multiagent robot teams,” in *First International Conference on Multi-Agen Systems (ICMAS)*, 1995.

[13] M. Roth, D. Vail, and M. Veloso, “A world model for multi-robot teams with communication,” in *IROS-2003*, 2003. (under submission).

[14] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[15] B. P. Gerkey, R. T. Vaughan, K. Stoy, A. Howard, G. S. Sukhatme, and M. J. Mataric, “Most valuable player: A robot device server for distributed control,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[16] R. A. Brooks, “Integrated systems based on behaviors,” *SIGART Bull.*, vol. 2, no. 4, pp. 46–50, 1991.

[17] B. B. Werger, “Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams,” *Artificial Intelligence*, vol. 110, pp. 293–320, 1999.

[18] D. W. Gage, “Command control for many–robot systems,” in *Proceedings of Nineteenth Annual AUVS Technical Symposium*, June 1992.

[19] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *6th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, June 2002.

[20] W. M. Spears and D. F. Gordon, “Using artificial physics to control agents,” in *Proceedings of IEEE International Conference on Information, Intelligence, and Systems*, 1999.

[21] D. F. Gordon, W. M. Spears, O. Sokolsky, and I. Lee, “Distributed spatial control, global monitoring and steering of mobile agents,” in *Proceedings of IEEE International Conference on Information, Intelligence, and Systems*, 1999.

[22] G. Baldassarre, S. Nolfi, and D. Parisi, “Evolving mobile robots able to display collective behaviors,” in *Proceedings of the International Workshop on Self-Organization and Evolution of Social Behaviors*, pp. 11–22, September 2002.

[23] E. Şahin and N. Franks, “Measurement of space: From ants to robots,” in *Proceedings of WGW 2002: EPSRC/BBSRC International Workshop Biologically-Inspired Robotics: The Legacy of W. Grey Walter*, (Bristol, UK), pp. 241–247, Aug. 14-16, 2002.

[24] S. Koenig and Y. Liu, “Terrain coverage with ant robots: A simulation study,” in *Proceedings of the International Conference on Autonomous Agents*, pp. 600–607, 2001.

[25] B. B. Werger and M. J. Mataric, “From insect to internet: Situated control for networked robot teams,” in *Annals of Mathematics and Aritificial Intelligence*, pp. 173–197, 2001.

[26] D. F. Gordon-Spears and W. M. Spears, “Analysis of a phase transition in a physics-based multiagent system,” in *Proceedings of NASA–Goddard/IEEE Workshop on Formal Approaches to Agent-Based Systems*, 2002.

[27] J. McLurkin and J. Smith, “Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots,” in *7th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, June 2004.

[28] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, “Flocking in fixed and switching networks,” *IEEE Transactions on Automatic Control (submitted)*, 2005.

[29] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, “Stable flocking of mobile agents, Part i: Fixed topology,” in *IEEE Conf. on Decision and Control*, 2003.

[30] J. P. Hespanha, “Extending LaSalle’s invariance principle to switched linear systems,” in *IEEE Conf. on Decision and Control*, 2001.

[31] J. P. Hespanha, D. Liberzon, A. S. Morse, B. D. O. Anderson, T. S. Brinsmead, and F. D. Bruyne, “Multiple model adaptive control, Part 2: Switching,” *International Journal of Robust and Nonlinear Control*, 2001.

[32] K. R. Gabriel and R. R. Sokal, “A new statistical approach to geographic variation analysis,” *Systematic Zoology*, vol. 18, pp. 259–278, 1969.

[33] D. W. Matula and R. R. Sokal, “Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane,” *Geographical Analysis*, vol. 12, pp. 205–222, 1980.

[34] J. W. Jaromczyk and G. T. Toussaint, “Relative neighborhood graphs and their relatives,” *Proceedings of the IEEE*, vol. 80, pp. 1502–1517, 1992.

[35] R. O. Saber and R. M. Murray, “Consensus protocols for networks of dynamic agents,” in *American Control Conference*, 2003.

[36] B. Shucker, T. Murphey, and J. K. Bennett, “Testbed implementation of wireless distributed control,” in *IEEE Conference on Robotics and Automation (submitted)*, 2007.

[37] M. Egerstedt, “On the specification complexity of linguistic control procedures,” *International Journal of Hybrid Systems*, vol. 2, pp. 129–140, March and June 2002.

[38] E. Klavins, R. Ghrist, and D. Lipsky, “A grammatical approach to self-organizing robotic systems,” *IEEE Transactions on Automatic Control*, vol. 51, pp. 949–962, June 2006.