Convergence Preserving Switching for Topology Dependent Decentralized Systems

Brian Shucker, *Member, IEEE,* Todd Murphey, *Member, IEEE,* and John K. Bennett, *Member, IEEE,*

***Abstract*— Stability analysis of decentralized control mechanisms for networked, coordinating systems has generally focused on specific controller implementations, such as nearest neighbor and other types of proximity graph control laws. This approach often misses the need for the addition of other control structures to improve global characteristics of the network. An example of such a situation is the use of a Gabriel graph, which is essentially a nearest neighbor rule modified to ensure global connectivity of the network if the agents are pairwise connected through their sensor inputs. We present a method of ensuring provable stability of decentralized switching systems by employing a hysteresis rule that uses a zero-sum consensus algorithm. We demonstrate the application of this result to several special cases, including nearest neighbor control laws, Gabriel graph rules, diffuse target tracking, and hierarchical, heterogeneous systems.**

I. Introduction and Related Work

Recent advances in integration and wireless communication have increased interest in the control problem associated with large numbers of cooperating agents. Although there is a significant body of previous work dealing with coordination of relatively small teams of agents, e.g. [1]–[5], large teams present new challenges. We are particularly interested in the problem of fully decentralized control (in some communities referred to as *swarming*), in which highly structured and potentially useful formations are created without any centralized coordination. Computation, communication bandwidth, and range constraints make effective decentralized algorithms necessary when the number of agents is large.

To address this challenge, there has been significant research into behavior-based and virtual-physics based control of large teams of agents [6]–[11]. Additionally, the control foundation of these systems has been explored extensively [12]–[16]. Each proposed system, of course, has its own advantages and drawbacks.

Many decentralized control algorithms are modeled after phenomena observed in nature, such as the flocking behavior of birds, or the schooling behavior of fish [17]. Others are based on simulated physical systems, such as cellular automata in crystals [18] or biological cells. Common to these approaches are simple local control laws implemented by each agent, and designed in such a way that desirable global behaviors emerge. Descriptions of such systems are given in [6] and [7], among others (e.g, [9], [10], [19]).

There are many approaches to formally show stability properties in a variety of switched and decentralized systems, e.g. [12]–[16], [20]–[24], but typically these proofs impose constraints on the dynamics of the system and the proximity graph. For example, the results in [21] apply only to a specific potential function on the unit-disk graph, and the results in [16] apply to another particular potential function on a Voronoi graph. The difficulty associated with these prior works is that the stability results leave little room for task specification; tasks must be framed in terms of what can be achieved in a stable manner and may therefore be 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. Finally, heuristics that are not easily combined with these approaches 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 mechanism should dictate task specification to the minimum extent possible. To this end, we have developed a more general method of proving convergence, focusing on ease of implementation and genericity of proximity graphs to which it is applicable.

One general method for proving stability for a control system is to find a Lyapunov function. This is simply a potential function that is always positive and decreasing, except at the desired stable point, where it is zero. If such a function exists, then over time, the system must evolve to the stable point. Finding a Lyapunov function becomes difficult, however, for hybrid systems that can switch between many states with differing dynamics. In fact, 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 [25]. This is often an intractable problem for systems with large numbers of hybrid states. Coordinated control systems are in this class of systems, having as many as $(n-1)!$ states for n agents.

Dwell-time analysis, such as described in [23], seeks to provide stability for a more general class of systems by imposing restrictions on the (global) switching rate. We base our work on these initial results. We extend the results to apply this type of analysis to decentralized systems, where each agent has access only to local information. In addition, we demonstrate how to use a consensus algorithm as part of the hysteresis-generating function in order to decentralize the approach.

Our work thus takes advantage of the intuition behind dwell-time analysis to produce a general technique for proving stability in the sense of convergence (like that used in linear systems) using only local information. For our purposes, we are interested only in stabilizing the kinetic energy of the system; that is, we prove that the agents converge to some particular state, but do not specify that state directly. Instead, the

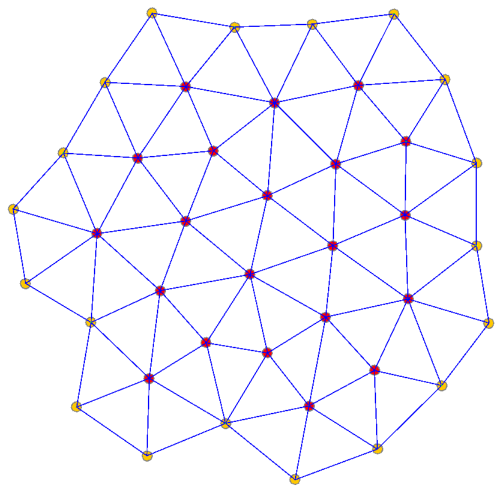


Fig. 1. Gabriel graph example: final formation after deploying initially from a tight cluster

final state is dictated by the properties of the specific system to which our technique is applied. Thus, we can take any instance of a large class of simple, perhaps heuristically-driven control systems, and transform it into a similar system that guarantees convergence to some configuration. The transformation takes the form of a straightforward modification of the switching function. In cases (such as area coverage) where configuration stability is not required, the simplicity and flexibility or our approach makes it easier to design a practical system.

Throughout this paper, the motivating example will be a switching function that produces a Gabriel graph, which we describe in more detail in [26] and [27]. This Gabriel graph switching function is described briefly in Section II. Sections III and IV detail the generalized result that applies to a large class of systems, including the motivating example. Sections V-A, V-B, and VI present additional examples of common situations where this technique may be put to use. Section VII addresses collision avoidance, and the implications of limited sensor range are discussed in Section VIII. We briefly discuss the impact of our technique on performance in Section IX and conclude in Section X.

II. MOTIVATING EXAMPLE: VIRTUAL PHYSICS SPRING MESH

In previous work [26], [27], we analyzed a decentralized control system employing a virtual physics model of a spring mesh. 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 i is

$$\begin{array}{c} \ddot{\mathbf{x}}_i = \mathbf{u}_i \\ \mathbf{u}_i = \left[\sum_{j \in N_i} k_s(\|\mathbf{x}_i - \mathbf{x}_j\| - l_0)\hat{\mathbf{v}}_{ij} \right] - k_d\dot{\mathbf{x}}_i \end{array} \tag{1}$$

where $\mathbf{x}_i$ represents the Cartesian coordinates describing the agent's position, $\ddot{\mathbf{x}}_i$ is the agent's acceleration, $\dot{\mathbf{x}}_i$ is the agent's velocity, N_i is the set of springs connected to this agent, and $\hat{\mathbf{v}}_{ij}$ is the unit vector from agent i to agent j. Control constants are the natural spring length (l_0), the spring stiffness (k_s), and the damping coefficient (k_d). 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 [26]). The standard Lyapunov approach simply requires us to find a potential function that is always positive and decreasing, except at the stable point (where it is zero). This is the primary motivation for using a virtual physics model–the virtual kinetic energy provides a natural Lyapunov function candidate even for very complex systems. However, it is often useful to allow the creation and destruction of springs. For example, when the proximity graph is changing dynamically over time, springs will be created and destroyed [26].

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* (also referred to as the *neighbor graph*) G_N 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. This graph is not static; rather, its edge set varies over time according to some *switching function*, which determines the state of the edge set at any particular time. To simplify notation, we will understand S to be the edge set of G_S and N to be the edge set of G_N. N (and therefore G_N) will be defined by a time-varying switching function σ, which we will describe in terms of a graph construction algorithm. Note that N is necessarily a subset of S.

In prior work [26], [27], we introduced a switching function that creates a Gabriel graph G_N [28]–[30]. G_N dictates which data is incorporated into the control laws. In particular, the switching function dictates N. 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 iff there are no other agents within the circle with diameter $\overline{AB}$. A simulated example of deployment using the Gabriel graph switching algorithm is shown in Figure 1.

The Gabriel graph switching function provides many advantages; chief among these is provable connectivity of the graph [29]. The Gabriel graph is also well-suited to providing uniform coverage of an area, as it creates a mesh of acute triangles. The Gabriel graph is a planar graph [29], so it does not suffer from high edge density when the agents are close together. However, the Gabriel graph depends on springs being created with non-zero virtual potential. This complicates any proof of stability, as virtual 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" [23]. 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 reserve to compensate for this increase.) We find this approach intuitive and more straightforward to implement in our decentralized system, in which switching events are detected locally. Although any global quantity can be problematic, we will demonstrate that a local estimate of this quantity based upon a zero sum consensus algorithm is sufficient to establish stability.

Consider a set of agents $r_i \in R$. Let the time-varying signal $\sigma(t)$ be the switching function for a Gabriel graph G_N (i.e., σ determines the time evolution of G_N). For convenience, we will denote this function as $\sigma : t \mapsto G_N$, as it takes t as an input to produce G_N as an output. 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 [26] 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 that the system is stable during the intervals between switches. This is due to Barbalat's lemma [31], 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 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 III. 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 $\mathbf{U}_i$.

$$\mathbf{U}_i = \sum_{h \in N_i} \frac{1}{2} k_s (\|\mathbf{x}_i - \mathbf{x}_h\| - l_0)^2$$

where N_i is the set of springs connected to agent i. 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:

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

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 relationship holds:

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

Additionally, let

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

where $\mathbf{x_i}$ represents the position of agent i. The quantity d_i represents the rate of virtual energy dissipated by damping at agent i. It is a direct consequence of the static stability proof in [26] that on any interval between switches, the following equality holds:

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

This result follows from the fact that the virtual physics is based on a spring mesh system, where all 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 virtual 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 IV). E_i is initialized to an arbitrary nonzero value. As virtual energy is damped out of the system, a fraction of that energy is added to the reserve. When a switch occurs, the virtual 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 an agent i may not create a spring if that operation would cause E_i to become less than zero. A more precise definition of $\sigma'(t)$ will be given in Section III.

Notice that preventing spring creation requires the cooperation of two agents (one on each end), since the properties of $\mathbf{V}_{\sigma(\tau_j)}$ given above depend upon symmetry in the springs (that is, G_N 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 [26]. 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 [26], of which the Gabriel graph is a member.

III. General Result

We will now present a formal proof that applies to our motivating example, as well as a large class of similar switched systems. Note that while this proof is clearly applicable to many systems that use interaction graphs as the basis for their switching function, other types of switching functions also have the required properties.

Consider a set of agents R and a time-varying switching signal σ that is constant except for discrete changes whenever a switch occurs. 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, $\sigma : (\mathbf{x}, t) \mapsto f$. We assume the following properties:

A1 For any time interval $(t_j...t_{j+1})$ on which σ is constant (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 the union of all functions $\mathbf{V}_{\sigma(\tau_j)}$.

A2 For all times t, $\lim_{\tilde{t}\to t^-} \dot{\mathbf{V}}_{\sigma(\tilde{t})} = \lim_{\tilde{t}\to t^+} \dot{\mathbf{V}}_{\sigma(\tilde{t})}$.

A3 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 $\dot{\mathbf{V}}_{\sigma(t)}$.

A4 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 $\sum_{i\in R} \hat{s}_i \geq \sum_{i\in R} s_i$.

A5 A switch at time t_j for which $\hat{s}_i > 0$ for any $i \in R$ may be prohibited. More precisely, the *nominal* switching function σ may be replaced in the control laws by a *modified* switching function $\sigma' : (\mathbf{x}, t) \mapsto f$ which behaves like σ, but with the added property that σ' may (or may not) delay or omit a switch for which $\hat{s}_i > 0$ for any $i \in R$. The modified switching function σ' is actually used for control rather than the nominal σ, which may be thought of as a reference switching function.

Property **A1** implies that the system is stable in the absence of switching. This is typically simple to verify using standard Lyapunov function techniques, and holds in the case of our motivating example.

Property **A2** allows switches to cause discrete changes in the potential of the system, but not in the damping rate. A simple way of ensuring that Property **A2** is true in practice is to define a switching function that cannot have any instantaneous effects on damping. This is done in our motivating example, where switches affect the amount of stored energy in springs, but not the kinetic energy of the agents, which is the term that controls damping.

Property **A3** involves the agents' local estimates of the amount of virtual energy that is being damped out of the system. If Property **A3** is satisfied, then the sum of the local estimates does not collectively over-estimate the amount of damping that occurs.

Our motivating example satisfies this property, because the total damping is equal to the sum of the damping at each individual agent–a quantity which is known precisely at the agent. However, only an upper bound is necessary; it may be beneficial to underestimate the damping in a practical implementation, in order to more easily ensure that the property is satisfied in the face of constraints such as imprecise sensing.

Property **A4** involves the agents' local estimates of the potential created by switching. If Property **A4** is satisfied, then the sum of the local estimates does not collectively under-estimate the actual potential created by a switch.

In our motivating example, the potential of each spring is known precisely. If each agent computes the change in potential caused by the creation and destruction of springs for which it is an endpoint, the sum of the local estimates will exactly equal the actual change in potential. Similarly to Property **A3**, a lower bound is sufficient, which should make practical implementation more straightforward.

Property **A5** tells us that the the *nominal* switching function σ (which is based on the sensor graph G_S and is typically designed a priori to satisfy network topology requirements–the Gabriel graph is just one example of such a graph) may be implemented using a *modified* switching function σ'. The modified switching function σ' is what is actually used in the generation of control laws for each agent. The modification of σ allows for separating the control design into two components; the low-level control architecture, and the high-level topology built upon the low-level stability properties. The meta-level controller that transforms σ into σ' uses only very limited sensing capabilities (e.g., proximity but not distance or ordering). Nevertheless, we will see that it is crucial to maintain stability.

It is important to note that the ability to prohibit a switch must only be satisfied in cases where $\hat{s}_i > 0$. Intuitively, this means that only switches that *increase* the potential in the system must be controllable. In our motivating example, it is possible for environmental conditions to cause a switch (as with the loss of a communication link, for example) that cannot be prevented. However, these uncontrollable switches have $\hat{s}_i \leq 0$ by design; the loss of a link can only *decrease* the system potential. If communication is re-established, the link is not necessarily added back into the control graph; thus, it is possible to control the switch in the positive direction. In general, it is necessary to define systems such that uncontrollable events cannot increase the overall potential.

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

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

$$\dot{E}_i(t) = -k_e d_i(t) \text{ if } s_i(t) = 0 \tag{2}$$
$$E_i(t) = \lim_{\tilde{t}\to t^-} E_i(\tilde{t}) - s_i(t) \text{ otherwise} \tag{3}$$

where k_e is a global constant, $0 < k_e < 1$. Notice that E_i is initialized to a nonnegative value and then evolves according to Equation 2 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 3.

Each agent maintains a local estimate $\hat{E}_i$, which is initially greater than zero and evolves as follows:

$$\dot{\hat{E}_i}(t) = -k_e d_i(t) \text{ if } \hat{s}_i(t) = 0 \tag{4}$$
$$\hat{E}_i(t) = \lim_{\tilde{t}\to t^-} \hat{E}(\tilde{t}) - \hat{s}_i(t) \text{ otherwise} \tag{5}$$

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

$$E = \sum_{i \in R} E_i \tag{6}$$

$$\hat{E} = \sum_{i \in R} \hat{E}_i \tag{7}$$

We will call E the *global energy reserve*.

This brings us to the simple change necessary to stabilize the system. The *modified switching function* $\sigma' : (\sigma, \mathbf{x}, t) \mapsto f$ is an identity on 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 **A5**. 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 **A3**). Also, this computation is decentralized; the agents only need access to the local values $\hat{E}_i$, d_i, and $\hat{s}_i$.

$$\sigma'(\mathbf{x}, t) = \begin{cases} \sigma(\mathbf{x}, t) & \text{if } E_i > 0 \text{ for all } i \\ \lim_{\tilde{t} \to t^-} \sigma'(\mathbf{x}, \tilde{t}) & \text{otherwise} \end{cases}$$

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 6 and 7 and the definitions of s_i and $\hat{s}_i$ (see Property **A4**) that $E \geq \hat{E}$. Thus if $\hat{E} \geq 0$, then $E \geq 0$ as well.

Theorem 3.1: In any system satisfying Assumptions **A1-A5** and using the modified switching function σ', all agents eventually reach a state of unchanging potential. That is, $|\mathbf{V}_{\sigma'(t)} - \alpha| \rightarrow 0$ for some $\alpha \in \mathbb{R}$ and, in particular, $\dot{\mathbf{V}}_{\sigma'(t)} \rightarrow 0$.

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:

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

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:

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

Substituting for $\dot{E}$:

$$\dot{\mathbf{V}}' = \dot{\mathbf{V}} + \sum_{i \in R} -k_e d_i \tag{8}$$

To handle switches, we must look back to the definition in Property **A4**:

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

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

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

Substituting for E from Equation 3,

$$\lim_{\tilde{t} \to t^+} \mathbf{V}'(\tilde{t}) = \lim_{\tilde{t} \to t^-} \mathbf{V}(\tilde{t}) + \sum_{i \in R} s_i(t) + \lim_{\tilde{t} \to t^-} E(\tilde{t}) - \sum_{i \in R} s_i(t) \tag{9}$$

which simplifies in the following way:

$$\begin{aligned} \lim_{\tilde{t} \to t^+} \mathbf{V}'(\tilde{t}) &= \lim_{\tilde{t} \to t^-} \mathbf{V}(\tilde{t}) + \lim_{\tilde{t} \to t^-} E(\tilde{t}) \\ &= \lim_{\tilde{t} \to t^-} \mathbf{V}'(\tilde{t}) \end{aligned} \tag{10}$$

Thus, the discontinuity in $\mathbf{V}'$ has been removed, as the limits from both sides are the same. We know from Property **A2** that switches do not cause discontinuities in $\dot{\mathbf{V}}'$ either, so Equation 8 holds true at all times.

Further, since $\dot{\mathbf{V}}$ is negative definite (from Property **A1**), $0 < k_e < 1$, and $\dot{\mathbf{V}} < \sum_{i \in R} k_e d_i < 0$ (from Property **A3**), 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 **A3**), 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 $\dot{\mathbf{V}}' \rightarrow 0$ as $t \rightarrow \infty$. It follows directly that $\dot{\mathbf{V}}_{\sigma'(t)} \rightarrow 0$ as $t \rightarrow \infty$.

■

Note that in the proof of Theorem 3.1 we are effectively changing both where the switch in σ is allowed to occur and potentially which switches are allowed to occur.

Remark 3.1: While our proof based on Barbalat's lemma is convenient for smooth potentials, it is not the only technique that is compatible with the energy reserve approach. For example, consider the work of Tanner et. al. in [21]. A control input u and Lyapunov function $\mathbf{V}$ are presented (we have changed the notation slightly to match the conventions used here):

$$u_i = -\sum_{j \in N_i} (\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j) - \sum_{j \in N_i} \nabla P(\mathbf{x}_i, \mathbf{x}_j)$$

where P is some potential function that approaches infinity as $\mathbf{x}_i$ approaches $\mathbf{x}_j$, and has a unique minimum when agents i and j are at a desired distance. N_i is the set of neighboring agents within some threshold distance of agent i. ∇P represents the gradient of P with respect to $\mathbf{x}$.

$$\mathbf{V} = \frac{1}{2} \sum_{i \in R} \left[\sum_{j \in N_i} P(\mathbf{x}_i, \mathbf{x}_j) + \dot{\mathbf{x}}_i^T \dot{\mathbf{x}}_i) \right]$$

$$\dot{\mathbf{V}} = \dot{\mathbf{x}}^T L \dot{\mathbf{x}}$$

where L is the Laplacian of the neighbor graph (the neighbor graph is defined by the union of N_i for all agents i).

It is simple to add an energy reserve to $\mathbf{V}$, with $d_i = \dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j$. This would modify the Lyapunov function as shown:

$$\mathbf{V} = \frac{1}{2}\sum_{i\in R}\big[\sum_{j\in N_i} P(\mathbf{x}_i,\mathbf{x}_j) + \dot{\mathbf{x}}_i^T\dot{\mathbf{x}}_i)\big] + E$$
$$\dot{\mathbf{V}} = (1-k_e)\dot{\mathbf{x}}^T L\dot{\mathbf{x}}$$

This change carries through the rest of the analysis. Similarly to our results, the results in [21] are preserved with the addition of an energy reserve, which allows for more flexibility in specifying a switching function.

Remark 3.2: In our motivating example, it is the case that the trajectories for all agents are defined for all t. This will also be the case for the example systems presented in Section V; however, it is possible to construct a system that meets the assumptions presented here and yet does not define the trajectories of all agents at all times. This can occur because switches between states of equal potential may happen arbitrarily fast, potentially resulting in an infinite number of switches.

In the event that a system cannot be designed to avoid such a situation, infinite switching may be prevented by adding a small cost to each switch. Precisely, replace Equation 3 with the following:

$$E_i(t) = \lim_{\tilde{t}\to t^-} E_i(\tilde{t}) - s_i(t) - \epsilon \tag{11}$$

where ϵ is a very small but finite constant. This modification ensures that the number of switches is finite, so existence of solution can be proven using existing methods (e.g., see [32]).

IV. Energy Reserve Consensus

Although the decision to prohibit a switch is made by each agent based on its local energy reserve, it may be 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 [33]. This algorithm allows a decentralized set of agents to reach a consensus on a common global value, while sharing information only with their local neighbors. We will apply this algorithm in a novel way, in order to combine local energy reserves into a single global reserve. If an agent i has a set of neighbors S_i that it can sense,

$$\bar{u}_i = \sum_{l\in S_i}(E_l - E_i) \tag{12}$$

We then replace Equations 2 and 4 with the following:

$$\dot{E}_i = -k_e d_i + \bar{u}_i \tag{13}$$
$$\dot{\hat{E}}_i = -k_e d_i + \bar{u}_i \tag{14}$$

Equations 3 and 5 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:

$$\sum_{i\in R}\bar{u}_i = 0 \tag{15}$$

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 8 remains unchanged. Since Equation 3 is also unchanged, the result in Equations 9 through 10 also stands as before. The system evolves somewhat differently, as the times when we must prohibit a switch have changed due to the differing *local* values of E, but the system meets all the conditions necessary for the proof in Section III because the *global* behavior of E still has the required properties. However, as described in [33], all of the local energy reserves will now converge to a single value.

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 15 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 4.1: In any system satisfying assumptions **A1-A5** where Eqs.(12)-(14) replace Eqs.2 and 4, all agents eventually reach a state of unchanging potential.

V. Examples

A. Nearest Neighbors and Gabriel Graphs

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. [34]), 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:

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i$$
$$\mathbf{u}_i = \big[\sum_{j\in N_i}\nabla P(\mathbf{x}_i,\mathbf{x}_j)\big] - k_d\dot{\mathbf{x}}_i \tag{16}$$

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.

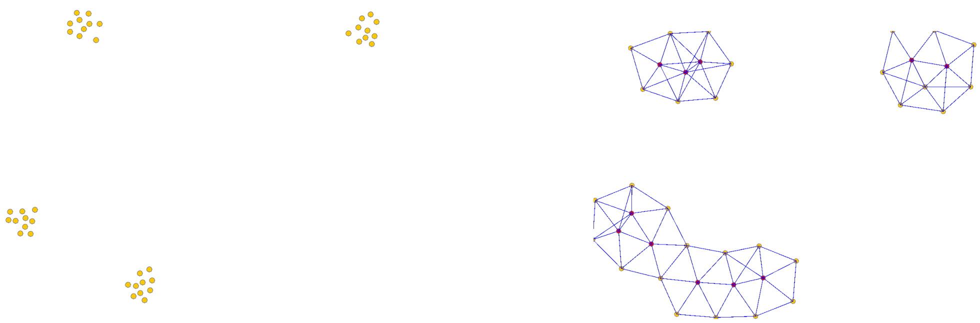
For each interval τ_j between switches, let the potential function be:

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

Since P is conservative, it can be shown that:

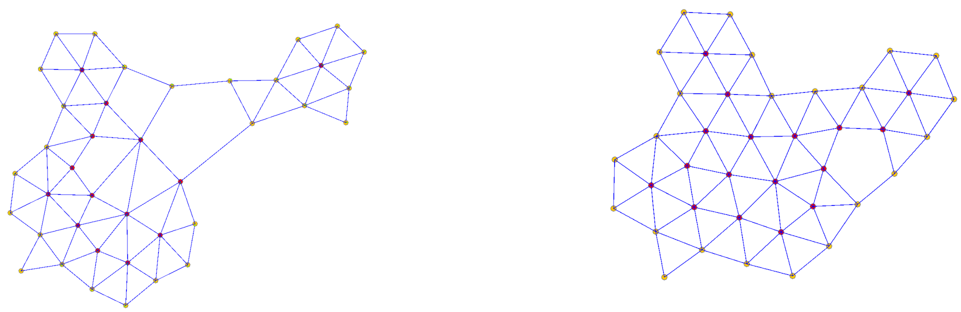
$$\dot{\mathbf{V}}_{\sigma(\tau_j)} = \sum_{i\in R} -k_d\dot{\mathbf{x}}_i^T\dot{\mathbf{x}}_i \tag{17}$$

Thus, this definition of $\mathbf{V}_{\sigma(\tau_j)}$ satisfies Property **A1**. Property **A2** is clearly satisfied, because the neighbor set does not appear in Equation 17.



(a) Initial state (b) Final state

Fig. 2. Nearby neighbors example: agents start in several separate clusters



(a) Intermediate state (b) Final state

Fig. 3. Gabriel graph switching function with initial conditions from Fig 2

The damping term makes it easy to satisfy Property **A3**; we simply let:

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

Since we can evaluate P at any point, satisfying Property **A4** is also straightforward. We define s_i such that:

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

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 **A5** 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 5.1: With the nearest neighbor graph topology from (16) where σ' is substituted as described in Section III, all agents eventually reach a state of unchanging potential.

Simulation results for such a system are shown in Figure 2. In this example, a simple spring potential identical to that used in Section II is used. However, any potential could be used, such as the one in [21].

For comparison, a simulation with the Gabriel graph switching function and the same initial conditions is shown in Figure 3. This shows the benefits of a switching function that has specific topological properties–in this case, the connectivity property of the Gabriel graph dictates that the agents will form a single connected graph so long as the associated sensor graph G_S is sufficiently connected.

B. 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 [9]), 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 virtual 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:

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

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

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

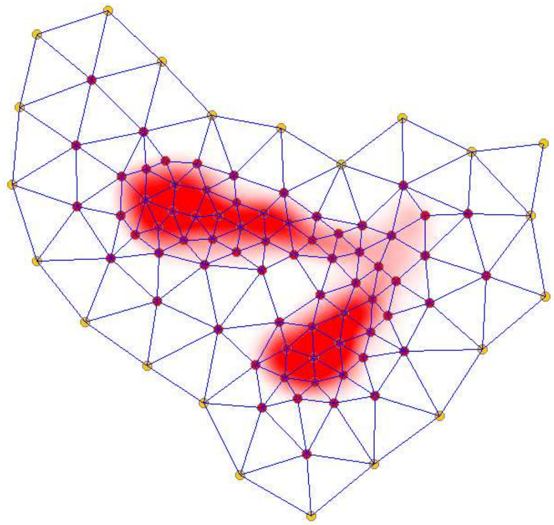


Fig. 4. Agents mapping a complex diffuse target

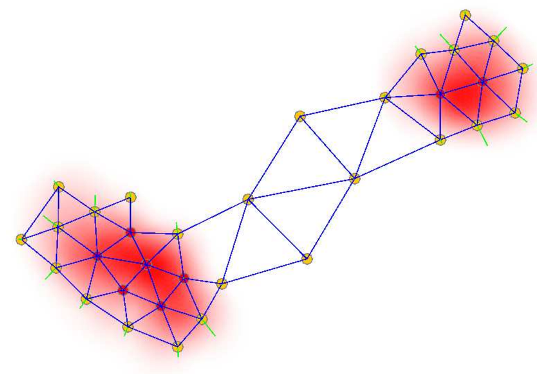


Fig. 5. Agents mapping multiple diffuse targets

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 III, Properties **A1** and **A2** are met because the potential functions are conservative and the system is damped independently of the targets (a proof of this is omitted for brevity, but is fairly straightforward). We can meet Property **A3** with the following (reasonable) choice:

$$d_i = -k_d \dot{\mathbf{x_i}}^{\mathbf{T}} \dot{\mathbf{x_i}}$$

Property **A4** 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 **A5** is met because $s_i > 0$ only when a target appears, and agents may elect not to track a given target if necessary.

Since each of the properties are satisfied, our technique may be applied. With the 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 4 and 5. 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 II, but with adaptive spring lengths so that the density of agents increases with the target intensity.

Figure 5 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. Explicit heterogeneity is considered in Section VI.

The combination of changing spring parameters and switching would normally be difficult to analyze. However, by applying the above 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.

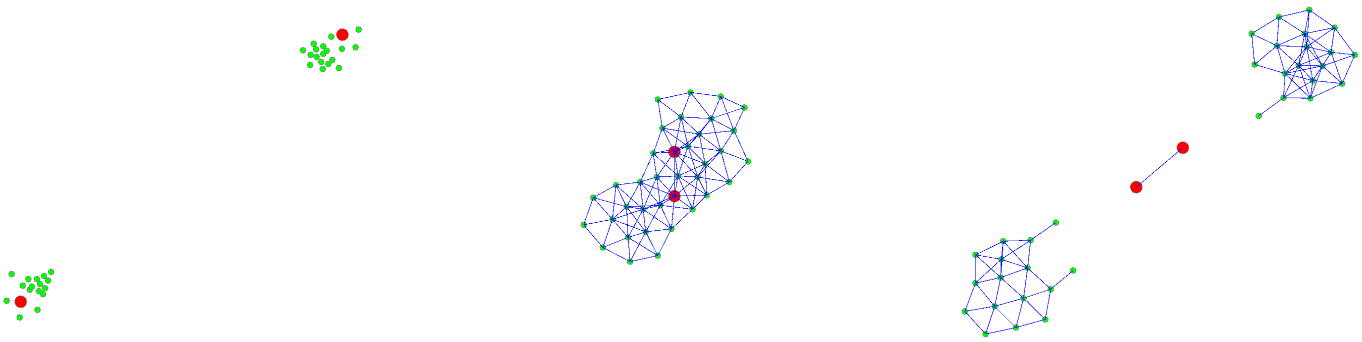
VI. Hierarchical Heterogeneous Systems

Suppose we have agents $r_i^l \in R_l$ (the lower hierarchy) and $r_i^u \in R_u$ (the upper hierarchy) with states $\mathbf{x}_i^l$ and $\mathbf{x}_k^u$ respectively. Accordingly, all agents in R_l are only equipped with short-range sensors and all agents in R_u are equipped with long-range sensors. Moreover, all agents in R_l employ a nearest neighbor (NN) control that includes all agents in both R_l and R_u, and all agents in R_u employ a Gabriel graph (GG) control law with other agents in R_u and a nearest neighbor control law with agents in R_l. We would like to know if the graph is connected (thereby utilizing the differences in sensing ability).

The key to using Theorem 3.1 is to maintain the symmetry of the connections. If we note the pairwise interactions by $(r_i^x, r_j^y) = T$ (where x and y are either u or l and T is either NN or GG), note that

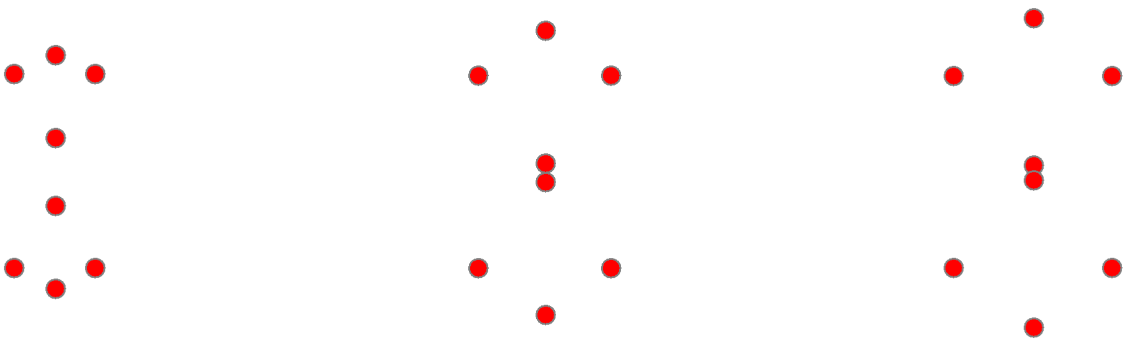
1 $(r_i^l, r_j^l) = NN$
2 $(r_i^l, r_j^u) = (r_i^u, r_j^l) = NN$
3 $(r_i^u, r_j^u) = GG$

so symmetry is maintained. For any particular static graph this is stable, so the resulting graph with switching from the NN and GG rules will be stable in the sense of Theorem 3.1, so long as each agent is using $\hat{E}$ and Eqs.(12)-(14) to monitor the switching. Simulations of this scenario are depicted in Figure 6. If the gain k_s^{NN} on the nearest neighbor control laws is much lower than the gain k_s^{GG} on the Gabriel graph control laws, then the system is still stable, but it can become disconnected, as shown in Figure 6. If, however, $k_s^{GG} < k_s^{NN}$, then the system stabilizes in a globally connected manner.



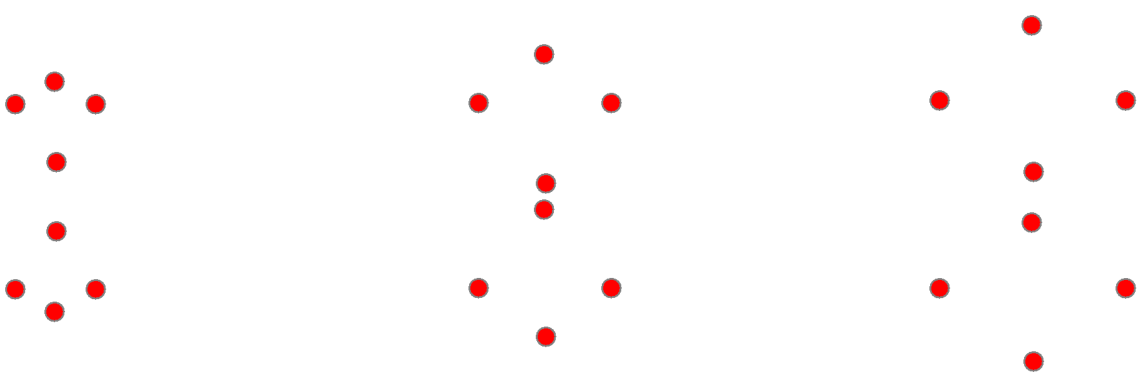
(a) Initial state (b) Final state, $k_s^{NN} > k_s^{GG}$ (c) Final state, $k_s^{NN} < k_s^{GG}$

Fig. 6. Heterogeneous agents, some with long-range sensors and some with short range sensors.



(a) snapshot 1 (b) snapshot 2 (c) snapshot 3

Fig. 7. Agents on collision course, with no collision term



(a) snapshot 1 (b) snapshot 2 (c) snapshot 3

Fig. 8. Agents on collision course, with collision term in effect

VII. 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. However, it is possible (and fairly straightforward) to design a system that does not depend on switching for collision avoidance. For example, consider the following control law:

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

$$\mathbf{u}_i = \big[\sum_{j\in N_i} \nabla P_1(\mathbf{x}_i, \mathbf{x}_j)\big] + \big[\sum_{k\in R} \nabla P_2(\mathbf{x}_i, \mathbf{x}_k)\big] - k_d\dot{\mathbf{x}}_i$$

where N_i is the set of neighbors according to some relation (such as a Gabriel graph), and R is the set of all agents. Suppose that P_1 and P_2 are both conservative functions, and that $P_2(\mathbf{x}_i, \mathbf{x}_k)$ approaches infinity as $\mathbf{x}_i$ approaches $\mathbf{x}_k$. It may be the case that P_2 is a "short-range" potential–it rapidly becomes small as the distance between the agents increases.

Similar to our previous examples, this system satisfies all of the requirements for Theorem 3.1. In addition, since P_2 affects all pairs of robots at all times, no collision can occur without overcoming an infinite potential. In a sense, there are two proximity graphs superimposed. P_1 represents the "normal" behavior of the system and involves switching, and P_2, which only acts over short range to prevent collisions, never switches.

Figures 7 and 8 show the effect of the collision-avoidance term P_2. In this case, P_1 is the simple spring-like potential used previously, N is defined by a Gabriel graph, and $P_2(\mathbf{x}_i, \mathbf{x}_j) = k_c(\|\mathbf{x}_i - \mathbf{x}_j\|)^{-1}$, where k_c is a constant. Initially, the upper and lower groups of agents are heading towards each other at high velocity, which puts the two middle robots on a collision course. Figure 7 shows the result without P_2, and Figure 8 shows the result with P_2 in effect.

It should be noted that some care must be taken to ensure that a collision-avoidance term does not cause unintended consequences. For example, a poorly-chosen control law may avoid collisions but allow undesired local minima in the potential function. While terms such as P_2 do not affect our ability to stabilize the system, they may alter its performance. In our example, P_2 was chosen carefully to be insignificant in the normal case (when collision is not imminent), so its effect on the system's behavior is minimal.

VIII. Limited Sensor Range

Due to the realities of limited sensing, it may be desirable to ignore interactions with agents beyond a certain threshold range. The ability to cut off interactions beyond a finite range, while still maintaining stability of the system, is a key advantage of the non-smooth potential function described by Tanner et. al. [21]. As mentioned in Remark 3.1, the results in [21] are compatible with an energy reserve, which makes it possible to introduce an additional type of switching limited by the energy reserve.

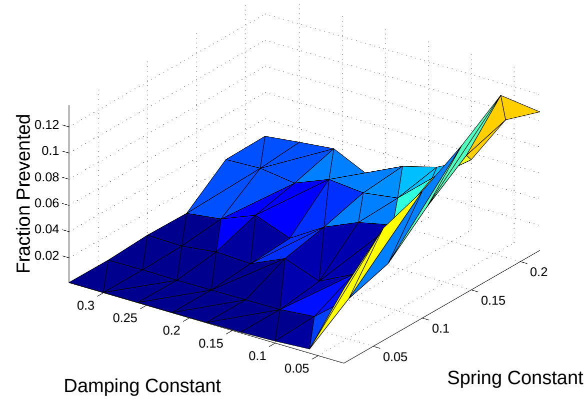


Fig. 9. Fraction of switches prevented

Consider the following system:

$$\begin{aligned}
\ddot{\mathbf{x}}_i &= \mathbf{u}_i \\
r_{ij} &= \|\mathbf{x}_i - \mathbf{x}_j\| \\
\mathbf{u}_i &= \big[\sum_{j\in R} \nabla P(r_{ij})\big] - k_d\dot{\mathbf{x}}_i \\
P(r_{ij}) &= \begin{cases} U(r_{ij}) & \text{if } r_{ij} < r_{max} \\ U(r_{max}) & \text{if } r_{ij} \geq r_{max} \end{cases} \\
U(r_{ij}) &= \frac{k_c}{r_{ij}^2} + \frac{1}{2}k_{ij}r_{ij}^2
\end{aligned}$$

where k_c is some positive constant, $k_{ij} = k_s$ if $j \in N_i$, and $k_{ij} = \epsilon$ otherwise, with $\epsilon << k_s$. N_i represents the neighbors of agent i according to some proximity graph, such as a Gabriel graph.

The potential function P meets the criteria for the non-smooth potential function described in [21]. However, the gain k_{ij} switches according to the proximity graph, becoming essentially zero and leaving only the collision-avoidance term when there is no corresponding edge in the graph. This switching is constrained by the energy reserve, preventing it from destabilizing the system. Thus, the stability result in [21] stands, the agents' behavior is dominantly controlled by a proximity graph that may be chosen arbitrarily, and the agents beyond a range of r_{max} may be ignored without impact on the system.

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 switches 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 9 shows the fraction of switches 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 switches are prevented. Only when the gain is relatively high and the damping constant is relatively low are there a large number of switches prevented. This result is highly intuitive–when we "push the envelope," with higher gains, we take greater risks with stability. The modified switching function comes into play more and more as we push the system towards higher performance.

X. 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 decentralized across a network of agents by additionally using consensus algorithms like those found in [33]. This leads to a flexible method of guaranteeing stability for arbitrary network graphs, and explicitly avoids instabilities due to the graph topology switching. Our technique is designed to be easily applicable to a wide range of systems, including those with heuristically-derived control laws, thus allowing formal proofs of stability to apply to many systems that are not addressed by existing methods.

We did not consider the effects of noise in this work, though it is largely addressed by the basic results of [23] on noise and external disturbances.

References

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

[2] 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.

[3] 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.

[4] M. Roth, D. Vail, and M. Veloso, "A world model for multi-robot teams with communication," in *IROS-2003*, 2003.

[5] 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.

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

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

[8] D. W. Gage, "Command control for many–robot systems," in *Proceedings of Nineteenth Annual AUVS Technical Symposium*, vol. 10 (4), pp. 28–34, June 1992.

[9] 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.

[10] 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.

[11] 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.

[12] M. Ji, S. Azuma, and M. Egerstedt, "Role-assignment in multi-agent coordination," *International Journal of Assistive Robotics and Mechatronics*, vol. 7, pp. 32–40, March 2006.

[13] R. A. Freeman, P. Yang, and K. M. Lynch, "Distributed estimation and control of swarm formation statistics," in *Proceedings ofthe American Control Conference (ACC)*, (Minneapolis, MN, USA), 2006.

[14] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[15] J. Cortes, "Characterizing robust coordination algorithms via proximity graphs and set-valued maps," in *Proceedings of the American Control Conference (ACC)*, (Minneapolis, MN, USA), pp. 8–13, 2006.

[16] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 2, pp. 1327–1332, 2004.

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

[18] 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.

[19] 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.

[20] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, pp. 988–1001, 2003.

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

[22] J. P. Hespanha, "Extending LaSalle's invariance principle to switched linear systems," in *IEEE Conf. on Decision and Control*, 2001.

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

[24] 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.

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

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

[27] 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.

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

[29] 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.

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

[31] I. Barbalat, "Systemes d'equations differentielles d'oscillations non lineaires," *Revue do Mathematiques Pures et Appliquees*, vol. IV, pp. 267–270, 1959.

[32] J. P. Hespanha, D. Liberzon, and A. S. Morse, "Bounds on the number of switchings with scale-independent hysteresis," in *IEEE Conf. on Decision and Control*, 2000.

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

[34] 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.