Technical Report CS-2012-03
Cohesion: Keeping Independently-Moving Agents Close Together

Luther A. Tychonievich and **James P. Cohoon**
Department of Computer Science
University of Virginia
Charlottesville, VA 22903

Abstract

Many algorithms for groups of moving physical agents require that the agents remain close enough together to coordinate cooperation. Most such algorithms do not provide facilities for maintaining inter-agent cohesion. Previous cohesion algorithms generally either supersede other mobility objectives (flocking) or constrain an entire collective to act as a single entity (formations). We present a framework for defining and maintaining agent cohesion without otherwise restricting agent behaviors. Our cohesion framework is designed as a drop-in filter on agent behaviors, preventing cohesion-breaking maneuvers without specifying particular behaviors for individual agents. The cohesion framework has similarity to many collision avoidance algorithms with respect to its applicability in conjunction with task-specific maneuvering algorithms. We prove that our framework guarantees cohesion and provide approximation techniques that ensure it is computationally tractable for broad classes of agents. We demonstrate the versatility of our approach by example.

1 Introduction

There is significant interest in decentralized artificial intelligence (AI) for cooperative mobile agents, also known as robot hives, swarms, or teams. A common assumption in these tasks is that the agents remain a single cohesive group. However, many agent behavior selection algorithms are not presented with a guarantee of cohesion. We have developed a technique for generating algorithms that guarantee cohesion. Our technique guarantees cohesion without imposing particular behaviors on individual agents. This separable guarantee allows agent behavior algorithm designers to ensure cohesion will be maintained without explicit algorithmic design to achieve that end.

Cooperating mobile agents offer several advantages over monolithic agents operating alone. Agent groups are more robust to individual failures, can sense and explore a larger region in parallel, and can work in more constrained spaces than can larger agents. Decentralization improves robustness and removes dependence on a central coordinating entity. From ant hives to economies to paired programming, there are countless examples of decentralized cooperating groups outperforming individuals.

While many decentralized AI algorithms assume cohesion, various approaches have been used to ensure continued cohesion. Flocking and formation algorithms maintain connectivity by dictating significant portions of each agent's actions, hampering or preventing individual behavior. Integrative approaches embed a guarantee of continued connectivity into the design of the task-specific algorithms the agents execute. None of these approaches provides a separable solution that guarantees the cohesion of the group without dictating other aspects of individual agent behavior.

We present a technique for constraining agent behaviors to preserve cohesion. These constraints prevent selection of behaviors that would result in agents separating from one another without specifying which cohesion-maintaining behavior each agent selects. These constraints are built around a distance metric based on possible future separation to ensure that past decisions never inhibit current cohesion. We provide tractable algorithms for evaluating the resulting predicates. We also present realizations of these techniques for several common mobile agent designs.

University of Virginia Department of Computer Science Technical Report #CS-2012-03.

Each element of our work is accompanied by proofs that show the preservation of connectivity among all agents.

2 Related Work

Many distributed AI algorithms assume that agents remain within contact of one another. Others utilize heuristic methods to (hopefully) reduce the likelihood of separation. Integrative approaches incorporate cohesion into the design of tasks-specific algorithms, a useful technique but one difficult to transfer to new problems.

Formations are a common approach for ensuring a collection of agents acts together. By specifying the relative location of each agent, formations ensure cohesion. Although some formations algorithms are decentralized, all effectively reduce the group to a single entity.

The quintessential cohesion algorithm is Reynold's Boids (Reynolds, 1987). Boids, like the many flocking, herding, and swarming algorithms that have followed it, provides a holistic model of agent behavior which guarantees a group remains connected in the absence of other maneuvering objectives (Yamins, 2005; Cucker and Smale, 2007). Many approaches have augmented flocking algorithms with additional objectives (Gurfil, 2005; Astengo-Noguez and Velzquez, 2008); to our knowledge, none provides revised guarantees of cohesion.

Two basic approaches have been used to guarantee cohesion with additional objectives. One approach is to switch from general unconstrained maneuvers to maneuvers that reduce separation when an agent gets to far from its fellows (Pereira, Das, and Kumar, 2003; Vazquez and Malcom, 2004; Hsieh et al.). Switching laws work well for highly maneuverable agents that can maintain cohesion with limited effort, but have not been shown to scale well to non-holonomic agents. The other leading approach is to design potential fields that integrate both cohesion and mission objective goals (Zavlanos and Pappas; Li et al., 2011). As with other integrative approaches, these are difficult to extend to new scenarios without breaking existing guarantees of cohesion.

Cohesion can be viewed as the dual of collision avoidance. Collision avoidance seeks to ensure for all possible behaviors of all neighboring agents at all future times that the separation between an agent and each neighbor is larger than some cutoff value. Cohesion tries to ensure the separation is smaller than a cutoff value instead. Many collision avoidance algorithms apply heuristics with some additional padding; these approaches are unsuited for the more constrained decision space of cohesion. Even non-heuristic collision avoidance algorithms generally consider only *expected* neighbor behaviors because there is often no solution to a universally-quantified collision avoidance. Cohesion, being cooperative, can instead consider all permissible neighbor behaviors.

Two collision avoidance techniques are of particular note. First, reciprocal velocity obstacles (van den Berg, Lin, and Manocha, 2008; Snape et al., 2011) are cooperative, using the fact that each agent is executing the same algorithm to streamline agent behavior. A similar notion of reciprocity is central to our cohesion algorithms. We utilize a more complicated model of reciprocity than the simple geometric version developed by van den Berg, Lin, and Manocha that applies to a broader class of agents; we also include a notion of liveness to prevent future separation. Second, generalized reactive navigation (Tychonievich, Burton, and Tychonievich, 2009) includes mechanisms for explicitly considering all possible neighbor behaviors and a generalized model of agent behavior. However, the over-approximations in that work cause deadlock in the more-constrained world of cohesion. We use a more precise approximation of agent behavior than Tychonievich, Burton, and Tychonievich and add a notion of liveness and reciprocity.

3 Generalized Agents and Cohesion

Although our principal motivation for this work is ensuring cohesion of mobile agents such as unmanned vehicles or digital avatars, our approach is not limited to a physical mobility interpretation. We present here a formal abstraction of agents and the cohesion objective.

An **agent** is defined by a **state** $s \in S$ and a set of achievable **behaviors** $\mathcal{B}$. A particular behavior $B \in \mathcal{B}$ defines an evolution of the state of the agent, which we could denote as a differential or difference equation (e.g., $\dot{s} = B(s)$) or as an evolution function (e.g. $s' = B(s, \Delta t)$). Each agent is assumed to be able to select its own behavior, possibly with a time delay before the new selection becomes active.

We use subscripts to denote the states obtained by following a behavior: B_s is defined such that $B_s(t)$ is the state an agent initially in state s would achieve by following behavior B for t seconds. Thus, $B_s(0)$ is s if B is defined at state s; note that not all behaviors need to be applicable at all states. We call the set of states for which a behavior is defined the **domain** of the behavior.

Definition 1 (Live Behaviors)**.** *A behavior B is **live for** s, where s is a state, if $B_s(t)$ is defined for all*

$t \geq 0$.

*A behavior is **live** if it is live for all states in its domain.*

*A behavior is **universal** if its domain is the set of all states. All universal behaviors are live.*

Definition 2 (Distance)**.** *A function $d : S \times S \to \mathbb{R}$ is called a **distance function** if, for all states a, b, and c: $d(a, a) = 0$, $d(a, b) = d(b, a)$, and $d(a, b) + d(b, c) \geq d(a, c)$.*

We speak informally of the notion of **position**. Two states s_1 and s_2 represent the same position if $d(s_1, s_2) = 0$. It may be possible to derive from S an inner product space whose induced norm defines the distance function. In such cases, the inner product space formalizes the notion of position.

If available, our algorithms can benefit from the existence of a unique **midpoint** between two agents.

Definition 3 (Midpoint)**.** *A state m is a **midpoint** between two other states a and b if $d(a, m) = d(b, m) = \frac{1}{2}d(a, b)$; it is a **unique midpoint** if, for any other midpoint m', $d(m, m') = 0$.*

Not all definitions of S and d will allow the definition of a unique midpoint. All terms including midpoints in our presentation are optional and may be ignored without loss of correctness.

Definition 4 (Connected)**.** *Two agents with states a and b are **connected** if the $d(a, b) \leq r$ for some fixed constant r.*

*A set of agents is **connected** if the graph with edges between connected pairs of agents is a connected graph.*

Definition 5 (Neighbors)**.** *Given three agents with states a, b, and c, the agent with state c is said to be **between** a and b if $d(a, c) < d(a, b) \wedge d(b, c) < d(a, b)$.*

*Two agents are **neighbors** if they are connected and no third agent is between them.*

Global cohesion requires that a set of connected agents remain connected in all future times. Locally, cohesion requires that each pair of neighbors remain connected until the interposition of a third agent removes their neighbor status.

Definition 6 (Local Cohesion)**.** ***Local cohesion** holds between two agents when being neighbors at time t implies being connected at time $t + \epsilon$ for all sufficiently-small ϵ.*

Local cohesion is a stronger constraint than global cohesion. For example, a ring of agents my break into a line while maintaining global, but not local, cohesion.

4 Constraints Providing Cohesion

We use the formalisms in the previous section to help specify a family of cohesion-preserving behaviors. The family is not complete; that is, there are cohesive behaviors not within this family. The incompleteness is offset by requiring only local computation (in both a spatial and temporal sense) with little or no inter-agent communication.

Our proofs of cohesion are inductively structured: locally cohesive behavior must not break connection between neighbors in the short term, and after the short term must be locally cohesive. The inductive step requires several intermediate tools, which we present first.

4.1 Null Trajectories and Halting Behavior

We define a set of "uninteresting" cohesive behaviors. Depending on the nature of the agents in question, these behaviors might correspond to standing still or having all agents move in lock-step.

Definition 7 (Null Trajectory)**.** *A set of behaviors $\mathcal{N}$ is called a **null trajectory** if both of the following hold:*

- *All $B \in \mathcal{N}$ are live (see Definition 1).*
- *The distance between any two agents following behaviors in $\mathcal{N}$ is monotonically non-increasing with t.*

Some null trajectories, like not moving, are trivial to define. Others, such as all moving along a common path, might require some kind of communication to coordinate. Communication required to agree upon a null trajectory is the only explicit communication used in our work.

Null trajectories are live but not necessarily universal. Allowing a restricted domain makes them easier to define, but also means they are not, in themselves, useful as fall-back behaviors when other cohesion techniques fail. The universal extension of a null trajectory is a halting behavior.

Definition 8 (Halting Behavior)**.** *A halting behavior H is a universal behavior (see Definition 1) such that, for any initial state, after finite time the behavior becomes the null trajectory. Formally, H is a halting behavior if*

$$\exists t^\star \geq 0 \text{ s.t. } \forall s \in S : H_{H_s(t^\star)} \in \mathcal{N}.$$

Universal null trajectories contain a halting behavior as a subset. Otherwise, halting behaviors perform the

maneuvering required to enter a state in the domain of the null trajectory.

We assume that, for any particular collection of agents, some halting behavior is defined. Several examples of halting behaviors are presented near the end of this document.

In general, it need not be the case that agents connected according to d have any behaviors in $\mathcal{B}$ that will preserve cohesion. For example, agents passing one another may not be able to turn around before they stray more than r apart. Therefore, we use a modified distance function to define connectivity.

$$d_H(s_0, s_1) \triangleq d(H_{s_0}, H_{s_1}).$$

By using d_H is lieu of d, we are guaranteed that H is always a cohesion-maintaining behavior. Thus there is a cohesion-maintaining option for any d_H-connected agents.

4.2 Constraints on Behavior

We now present a predicate that ensures a particular behavior is consistent with local cohesion. To do so, we first establish minimal assumptions about agent behavior as well as some supporting notation.

We assume a t_0 such that at any time t each agent is guaranteed to be able to select a new behavior by time $t + t_0$.

We introduce the following notation for constructing behaviors out of individual pieces:

Definition 9. *Let the notation $t_0[^A_B]$, where A and B are behaviors, represent the piecewise-defined behavior*

$$t_0[^A_B]_s(t) = \begin{cases} A_s(t) & 0 \le t \le t_0 \\ B_{A_s(t_0)}(t - t_0) & t_0 < t. \end{cases}$$

We define a conservative approximation of local cohesion in terms of three predicates:

Adversarial: The agent remains near its neighbor, no matter what its neighbor does.

$$\forall t \ge 0, B \in \mathcal{B} \quad d\Big(t_0[^A_H]_a(t), t_0[^B_H]_n(t)\Big) \le r. \quad (1)$$

Halt-like: The agent remains as close to its neighbor as it would have had it chosen the halting behavior.

$$\forall t \ge 0, B \in \mathcal{B}$$
$$d\Big(t_0[^A_H]_a(t), t_0[^B_H]_n(t)\Big) \le d\Big(H_a(t), t_0[^B_H]_n(t)\Big). \quad (2)$$

Midpoint: The agent remains within $\frac{r}{2}$ of the midpoint of the halting behaviors of it and its neighbor.

$$\forall t \ge 0 \quad d\Big(t_0[^A_H]_a(t), m\big(H_a(t), H_n(t)\big)\Big) \le \frac{r}{2}. \quad (3)$$

Definition 10. *The **local cohesion constraint** for a candidate behavior A of an agent in state a relative to its neighbor in state n is a combination of* (1), (2), *and, if a unique midpoint is defined,* (3)*:*

$$\begin{aligned} &\forall t \ge 0, B \in \mathcal{B} \\ &\quad d\Big(t_0[^A_H]_a(t), t_0[^B_H]_n(t)\Big) \le r \\ &\vee\ d\Big(t_0[^A_H]_a(t), t_0[^B_H]_n(t)\Big) \le d\Big(H_a(t), t_0[^B_H]_n(t)\Big) \\ &\vee\ d\Big(t_0[^A_H]_a(t), m\big(H_a(t), H_n(t)\big)\Big) \le \tfrac{1}{2}r \end{aligned} \quad (4)$$

Lemma 1.1. *Any pair of d_H-connected agents choosing behaviors that satisfy* (4) *will remain connected over $t \in [0, t_0]$.*

Proof. We consider each possible pairing of the predicates each agent might satisfy.

Either satisfies Adversarial: Connection is maintained by definition of the predicate no matter what the other agent is doing.

Both satisfy Halt-like: Since each agent is as close to the other as they would be were they halting, and since d_H-connection implies they would be connected if they were halting, then they must be connected.

Both satisfy Midpoint: Since m is unique and both agents are staying within $\frac{r}{2}$ of it, by the triangle inequality they are also within r of each other.

Midpoint and Halt-like: Call the agents satisfying Midpoint A_m and Halt-like A_h. A_m stays within $\frac{r}{2}$ of the midpoint m, and hence within r of H_{A_h}. A_h stays no farther from A_m than does H_{A_h}, so $d(A_m, A_h) \le r$.

□

Theorem 1 (Local Cohesion)**.** *If each agents always selects a behavior that satisfies* (4) *for each of its d_H-neighbors, then all agents maintain local d_H-cohesion. Additionally, selecting such behaviors is always possible.*

Proof. The halting behavior is defined everywhere (see Definition 8), and satisfies (2) (the second term of the constraint), meaning a satisfying behavior is always available.

By Lemma 1, we know that each pair of neighboring agents will remain connected over the immediate time window. Because a satisfying behavior is always available, each pair of neighboring agents will also remain connected over the subsequent time window. Inductively, local cohesion will hold for all future time. □

5 Behavior Validation

Previous discussion did not impose any constraints on the mathematical form of agent state and behaviors. To obtain a computational realization of the local cohesion constraint we impose a minimal structure on these parameters.

We consider behavior validation in the following discussion. We envision validation being combined with a task-specific selection algorithm, either by the selection algorithm generating candidate behaviors until one is validated or by the addition of a separate search function that seeks the "best" validated behavior.

In general, we do not expect d_H and the conditions of the local cohesion constraint to be tractably computable in closed form. Where they cannot be computed efficiently enough for effective behavior selection we compute tight but conservative approximations of the constraint. If we cannot prove that a behavior is safe, we reject that behavior.

We now consider a computational approach to solving (4). This approach first symbolically reduces the constraint to a system of polynomial inequalities and then uses a numerical approach to solve that system.

5.1 Polynomial Approximation

Our computational approach takes as input a quantified boolean expression of bounded-domain polynomial inequalities. We outline how arbitrary piecewise-smooth functions can be approximated as polynomial inequalities.

piecewise expressions including addition, scaling, and rational exponents of real-valued variables can be converted to polynomial inequalities in closed form. Rational exponents can be removed by integer powers and the possible addition of new terms; for example $a \leq \sqrt{b}$ becomes $a^2 \leq b \vee a < 0$. Piecewise expressions are re-written as disjunctions and expanded; for example,

$$\begin{cases} f_1 & g_1 \\ f_2 & \neg g_1 \end{cases} \leq \begin{cases} p_1 & q_1 \\ p_2 & \neg q_1 \end{cases}$$

is re-written as

$$(\neg g_1 \vee \neg q_1 \vee f_1 \leq p_1) \wedge (g_1 \vee \neg q_1 \vee f_2 \leq p_1) \\ \wedge (\neg g_1 \vee q_1 \vee f_1 \leq p_2) \wedge (g_1 \vee q_1 \vee f_2 \leq p_2).$$

Absolute values, sign functions, and other piecewise functions can be handled similarly.

For functions that include other operators, such trigonometric functions, we utilize polynomial approximation. It is well known that polynomials can be used to approximate any smooth function to any arbitrary level of precision over any finite interval, and that the error of such an approximation is no more than $\frac{c^n}{(n+1)!}$, where n is the order of the polynomial and c is proportional to the roughness of the approximated function. Algorithms and convergence analysis can be found in any good approximation text (e.g., Sederberg (2012) pp. 111–114).

5.2 Quantified Polynomial Inequalities

Boolean expression of multivariate polynomials under universal and existential qualifiers have been investigated at length by feedback control researchers. The overview of computational approaches performed by Dorato et al. indicates that the Bernstein branch-and-bound (BBB) method is best suited for our needs (Dorato et al., 2000).

The basic steps of the BBB method are:

1. Rearrange the inequalities to each contain a single polynomial, $P \leq 0$.
2. Rewrite each polynomial in the Bernstein basis, also called Bézier form.
3. The Bernstein coefficients form conservative bounds on the value of the polynomial; the polynomial also interpolates the corner coefficients. Since our inequalities are all universally quantified, an inequality is satisfied if all coefficients are are nonpositive, violated if all any corner coefficients are positive, and otherwise undetermined.
4. If the logical value of the boolean expression is undetermined, then one or more variables' domains can be split using de Casteljau's alogrithm. Each split portion must be satisfied for the original expression to be satisfied.

Algorithms for conversion to Bernstein basis and de Casteljau's algorithm can be found in any text on Bernstein polynomials (e.g., (Sederberg, 2012)).

Although the BBB method is numerically stable and converges well for well-conditioned polynomials, there are polynomials for which it converges poorly. All of these poorly-convergent polynomials are "close to" unsatisfied, in that some a small perturbation of coefficients will yield an unsatisfied expression. If the BBB method does not converge we halt it and conservatively interpret the constraint as unsatisfied.

6 Example Realizations

We present three example algorithms created using our technique for different definitions of agents.

Example 1 (Holonomic agent)**.** A **holonomic** agent is one that may freely move in any direction. First-order holonomic agents in a Euclidean environment can be modeled with state as position ($\vec{x}$) and a behavior as a velocity vector $\vec{v} = \frac{\partial}{\partial t}\vec{x}$, bounded by $\|\vec{v}\| \leq s'$, for some maximum speed s'. In this case, a reasonable null trajectory might be $\vec{v} = \vec{0}$. Since $\mathcal{N}$ is universal it is also a halting behavior.

We have the following formula for $t_0[{}^{\vec{v}}_{H}]_{\vec{x}}$:

$$t_0[{}^{\vec{v}}_{H}]_{\vec{x}} = \vec{p} = \begin{cases} \vec{x} + \vec{v}t & 0 \leq t < t_0 \\ \vec{x} + \vec{v}t_0 & t \geq t_0 \end{cases}, \quad \epsilon = 0$$

The second piece ($t \geq t_0$) doesn't impact the distance, since both agents are halting at that time. The equations we need to handle are:

$$H_{\vec{x}} = \vec{x}$$
$$t_0[{}^{\vec{v}}_{H}] = \vec{x} + \vec{v}t_0$$
$$d(\vec{x}_0, \vec{x}_1) = d_H(\vec{x}_0, \vec{x}_1) = \left\|\vec{x}_0 - \vec{x}_1\right\|_2$$
$$d\Big(t_0[{}^{\vec{v}_a}_{H}]_{\vec{x}_a}, t_0[{}^{\vec{v}_b}_{H}]_{\vec{x}_b}\Big) = \max_{0 \leq t \leq t_0} \left\|\big(\vec{x}_a - \vec{x}_b\big) + \big(\vec{v}_a - \vec{v}_b\big)t\right\|_2$$

Because $\|\vec{x}\|_2 = \sqrt{\vec{x} \cdot \vec{x}}$, we can convert the local cohesion constraint into an expression of polynomial inequalities as outlined above. For completeness, the conversion results in the following predicate. Given an agent with state $\vec{x}_a$ and a potential behavior $\vec{v}_a$, we evaluate the following for each neighbor with state $\vec{x}_n$:

$$\forall t \in [0, t_0], v_{ni} \in [-s', s']$$
$$s'^2 - \sum_i {v_{ni}}^2 \leq 0$$
$$\vee \ (1):$$
$$\sum_i \big(x_{ai} - x_{ni} + (v_{ai} - v_{ni})t\big)^2 - r^2 \leq 0$$
$$\vee \ (2):$$
$$\sum_i \big(x_{ai} - x_{ni} + (v_{ai} - v_{ni})t\big)^2 - \sum_i \big(x_{ai} - x_{ni} - v_{ni}t\big)^2 \leq 0$$
$$\vee \ (3):$$
$$\sum_i \left(x_{ai} + v_{ai}t - \frac{x_{ai} + x_{ni}}{2}\right)^2 - \frac{1}{4}r^2 \leq 0.$$

If the above expression evaluates to true for each neighbor then the agent can safely use $\vec{v}_a$ for at least the next t_0 seconds.

Example 2 (Accelerating agent)**.** Second-order holonomic agents control acceleration. A second order agent's state is position and velocity $(\vec{x}, \vec{v})$; a behavior is an acceleration vector $\vec{a} = \frac{\partial}{\partial t}\vec{v} = \frac{1}{2}\frac{\partial^2}{\partial t^2}\vec{x}$, bounded by $\|\vec{a}\| \leq a'$ for some maximum acceleration a'. We use as a null trajectory a constant velocity $\vec{v}_\emptyset$. A corresponding halting behavior is to accelerate toward $\vec{v}_\emptyset$.

We have the following formula for $t_0[{}^{\vec{a}}_{H}]_{\vec{x},\vec{v}}(t)$:

$$\vec{a}_H = \frac{(\vec{v}_\emptyset - \vec{v} - \vec{a}t_0)a'}{\|\vec{v}_\emptyset - \vec{v} - \vec{a}t_0\|_2}$$
$$t_a = \frac{1}{a'}\|\vec{v}_\emptyset - \vec{v} - \vec{a}t_0\|_2 + t_0$$
$$\vec{v}(t) = \begin{cases} \vec{v} + \vec{a}t & t \leq t_0 \\ \vec{v}(t_0) + (t - t_0)\vec{a}_H & t_0 < t \leq t_a \\ \vec{v}_\emptyset & t_a < t \end{cases}$$
$$\vec{x}(t) = \begin{cases} \vec{x} + \vec{v}t + \frac{1}{2}\vec{a}t^2 & t \leq t_0 \\ \vec{x} + \vec{v}_0 t + \frac{1}{2}\vec{a}t_0^2 + \frac{1}{2}(t - t_0)^2\vec{a}_H & t_0 < t \leq t_a \\ \vec{x}(t_a) + (t - t_a)\vec{v}_\emptyset & t_a < t \end{cases}$$

While $\vec{v}(t)$ is part of the state of the agent, only $\vec{x}(t)$ figures in the distance computation. Because the expression $d(t_0[{}^{A}_{H}]_a, t_0[{}^{B}_{H}]_b)$ contains two piecewise functions, each with three pieces, we split each inequality in the local cohesion constraint into nine pairings of pieces with appropriate guards. We then remove square roots and remove clauses that are trivially true or trivially false before proceeding with the BBB method. Although these steps are simple to follow, they require significant space and are not included here.

Example 3 (Car-like agent)**.** Car-like agents are traditionally modeled as non-holonomic, unable to move sideways. A car-like agent's state is position, heading, and speed $(\vec{x}, \theta, s)$; its behavior is a forward acceleration a, bounded by $|a| \leq a'$, and a signed curvature c, bounded by $|c| \leq c'$.

Agent motion is defined by the following definite integrals:

$$s(t) = s_0 + \int_0^t a(\tau)d\tau$$
$$\theta(t) = \theta_0 + \int_0^t c(\tau)s(\tau)d\tau$$
$$\vec{x}(t) = \vec{x}_0 + \int_0^t s(\tau)\vec{f}\big(\theta(\tau)\big)d\tau,$$

with "forward" vector $\vec{f}(\theta) \triangleq (\cos(\theta), \sin(\theta))$.

We assume the null trajectory is a fixed heading $\theta_\emptyset$ and speed $s_\emptyset$, and use a halting behavior that independently turns to that heading and accelerates to that speed as quickly as possible. Since orientation changes more rapidly at higher speeds, this halting behavior might not be optimal. However, as long as $s_\emptyset \neq 0$, it does meet the requirements of a halting behavior.

We assume that we can distinguish between θ and $\theta + 2\pi$. This assumption allows us to ignore the discontinuity in the halting behavior at $\theta_\emptyset \pm \pi$. We could handle that discontinuity explicitly by adding additional pieces to each function; we do not do so here to streamline our presentation.

We derive $t_0[_H^{a,c}]_{\theta_0,s_0}$ in several steps, starting an expression for speed:

$$
\begin{aligned}
a_H &= \text{sign}(s_\emptyset - s_0 - at_0)a' \\
t_a &= \frac{1}{a'}|s_\emptyset - s_0 - at_0| + t_0 \\
a(t) &= \begin{cases} a & t \le t_0 \\ a_H & t_0 < t \le t_a \\ 0 & t_a < t \end{cases} \\
s(t) &= \begin{cases} s_0 + at & t \le t_0 \\ s(t_0) + (t - t_0)a_H & t_0 < t \le t_a \\ s_\emptyset & t_a < t \end{cases}
\end{aligned}
$$

Since the rate of rotation depends on both speed and curvature, we derive several intermediary equations. The orientation of the agent while it is both accelerating and turning is θ_{ah}. It stops turning at time t_1 if it stops turning before it stops accelerating; otherwise it stops turning at time t_2.

$$
\begin{aligned}
\theta(t_0) &= \theta_0 + cs_0t_0 + \frac{1}{2}cat_0^2 \\
c_H &= \text{sign}(\theta_\emptyset - \theta(t_0))c' \\
\theta_{a_h}(t) &= \theta(t_0) + c_H s(t_0)(t - t_0) + \frac{1}{2}a_h c_H (t - t_0)^2 \\
t_1 &= \frac{-c_H s(t_0) + \sqrt{c_H^2 s(t_0)^2 - 2a_h c_H \left(\theta(t_0) - \theta_\emptyset\right)}}{c_H a_h} + t_0 \\
t_2 &= t_1 + \frac{\theta_\emptyset - \theta_{a_h}(t_a)}{s_\emptyset c_H} \\
t_c &= \max(t_1, t_2) \\
c(t) &= \begin{cases} c & t < t_0 \\ c_H & t_0 \le t < t_c \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

There are two cases for θ depending on the relative order of t_c and t_s, but both may be expressed in the same piecewise equation because the third piece only applies when $t_a < t_c$.

$$
\theta(t) = \begin{cases} \theta_0 + cs_0t + \frac{1}{2}cat^2 & t \le t_0 \\ \theta_{a_h}(t) & t_0 < t \le \min(t_a, t_c) \\ \theta_\emptyset + s_\emptyset c_H (t - t_c) & t_a < t \le t_c \\ \theta_\emptyset & \text{otherwise} \end{cases}
$$

The resulting formula for $\vec{x}$ is the integral of $s(\tau)\vec{f}\big(\theta(\tau)\big)$, which is a five-piece expression:

$$
\begin{cases} \big(s_0 + a\tau\big)\vec{f}\left(\theta_0 + cs_0\tau + \frac{1}{2}ca\tau^2\right) & t \le t_0 \\ \big(s(t_0) + (\tau - t_0)a_h\big)\vec{f}\big(\theta_{a_h}(\tau)\big) & t_0 < t \le \min(t_a, t_c) \\ (s_\emptyset)\vec{f}\big(\theta_\emptyset + s_\emptyset c_H(\tau - t_c)\big) & t_a < t \le t_c \\ \big(s(t_0) + (\tau - t_0)a_h\big)\vec{f}(\theta_\emptyset) & t_c < t \le t_a \\ (s_\emptyset)\vec{f}(\theta_\emptyset) & \max(t_a, t_c) < t \end{cases}
$$

Because this expression cannot be integrated in closed form, we apply a polynomial approximation of $\vec{f}$ to obtain a piecewise-polynomial integral that we can solve directly. The error terms are also polynomial. For example, the error in the first piece is $(s_0\varepsilon t + \frac{1}{2}\varepsilon a t^2)$, where ε is the (constant) error of the polynomial approximation. We add these errors into the inequalities (1), (2), and (3) to obtain conservative approximations. We then split the piecewise functions, square roots (in t_1), and sign dependence (in a_H, t_a, and c_H) to obtain a quantified boolean expression of polynomial inequalities.

7 Uncertainty and Asynchrony

Thus far we have assumed precise knowledge of the location of each agent and that all the agents make periodic decisions simultaneously. We made these assumptions in a conscious effort to keep the presentation simple, but they are not required for our method to work.

Any bounded uncertainty, be it in t_0, d, or any other constant, variable, or function we have discussed, can be incorporated into our model by introducing a new variable with the appropriate domain. For example, if t_0 is uncertain then we might write (1) as

$$
\forall t \ge 0, B \in \mathcal{B}, 0 \le t_a \le t_0, 0 \le t_n \le t_0 \\
d\Big(t_a[_H^A]_a(t), t_n[_H^B]_n(t)\Big) \le r.
$$

Uncertainty effecting estimates of the current distance between agents should also be interpreted conservatively to select the maximal number of possible neighbors.

Uncertainty need not be constant. For example, the error expression for the position of an agent with imperfect actuators may increase as a function of distance traveled. As long as the uncertainty can be expressed in terms of piecewise-smooth functions over bounded-domain variables, none of our presentation need change to accommodate it.

There is one class of uncertainty that can render us unable to guarantee cohesion. If agents are unable to reliably reach a null trajectory even with appropriate error terms on the distance function then no halting behavior is possible and none of our guarantees hold.

8 Conclusion and Future Work

We have presented a framework for guaranteeing that mobile agents remain within a fixed distance of one another without requiring any particular model of agent behavior and without constraining each agent's behavior beyond the cohesion constraint. We have also shown how this framework can be realized for three common agent designs.

The most obvious extension of our work is to define appropriate functions for other types of agents and environments, including finding halting behaviors for environments with static obstacles. We could also extend the technique to handle other maneuverability constraints, such as collision avoidance. Such an extension would likely resemble the reciprocal velocity obstacle technique (van den Berg, Lin, and Manocha, 2008) extended to arbitrary maneuvers and uncertainty in a manner similar to the generalized reactive navigation method (Tychonievich, Burton, and Tychonievich, 2009).

We have presented techniques guaranteeing local cohesion. While local cohesion guarantees global cohesion, it is a tighter restriction and can result in the group of agents getting stuck surrounding obstacles in the environment. A more general solution to global cohesion requires inter-agent consensus and is a subject of ongoing research.

We have presented our processes at a fairly high level. For implementation, we note that almost every element of our design is highly parallel: we could evaluate many potential behaviors across many neighbors simultaneously, even evaluating each linear inequality in parallel.

We have taken a problem that is traditionally handled either by heuristics or by significant over-constraint and have developed an approach for resolving the problem directly with general, proof-supported algorithms. We expect that this approach to agent behavior will be as beneficial to other researchers as are the cohesion-guaranteeing algorithms we have developed.

References

Astengo-Noguez, C., and Velzquez, L. 2008. A vectorial approach on flock traffic navigation. In *Artificial Intelligence, 2008. MICAI '08. Seventh Mexican International Conference on*, 300 –304.

Cucker, F., and Smale, S. 2007. Emergent behavior in flocks. *Automatic Control, IEEE Transactions on* 52(5):852 –862.

Dorato, P.; Li, K.; Kosmatopoulos, E.; Ioannou, P.; and Ryaciotaki-Boussalis, H. 2000. Quantified multivariate polynomial inequalities. The mathematics of practical control design problems. *Control Systems, IEEE* 20(5):48–58.

Gurfil, P. 2005. Evaluating uav flock mission performance using dudek's taxonomy. In *American Control Conference, 2005. Proceedings of the 2005*, 4679 – 4684 vol. 7.

Hsieh, M. A.; Cowley, A.; Kumar, V.; and Taylor, C. J. Maintaining network connectivity and performance in robot teams. *Journal of Field Robotics*.

Li, X.; Su, D.; Yang, J.; and Liu, S. 2011. Connectivity constrained multirobot navigation with considering physical size of robots. In *Proceedings of the International Conference on Automation and Logistics*, 24–29.

Pereira, G. A. S.; Das, A. K.; and Kumar, V. 2003. Decentralized motion planning for multiple robots subject to sensing and communication constraints. In *Proceedings of the 2003 International Workshop on Multi-Robot Systems*, 267–278.

Reynolds, C. W. 1987. Flocks, herds and schools: A distributed behavioral model. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 25–34. New York, NY, USA: ACM.

Sederberg, T. W. 2012. *Computer Aided Geometric Design*. Brigham Young University. `http://hdl.lib.byu.edu/1877/2822`.

Snape, J.; van den Berg, J.; Guy, S.; and Manocha, D. 2011. The hybrid reciprocal velocity obstacle. *Robotics, IEEE Transactions on* 27(4):696 –706.

Tychonievich, L. A.; Burton, R. P.; and Tychonievich, L. P. 2009. Versatile reactive navigation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009 (IROS 2009)*, 2966–2972. St. Louis, MO: IEEE.

van den Berg, J.; Lin, M.; and Manocha, D. 2008. Reciprocal velocity obstacles for real-time multiagent navigation. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 1928 –1935.

Vazquez, J., and Malcom, C. 2004. Distributed multirobot exploration maintaining a mobile network. In *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, volume 3, 113–118.

Yamins, D. 2005. Towards a theory of "local to global" in distributed multi-agent systems (ii). In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 191–198. New York, NY, USA: ACM.

Zavlanos, M. M., and Pappas, G. J. Distributed connectivity control of mobile networks.