Distributed Autonomous Systems
(Benchmark Proposal)

Hoang-Dung Tran, Luan Viet Nguyen, Weiming Xiang and Taylor T. Johnson

Vanderbilt Universitsy, Tennessee, USA
taylor.johnson@gmail.com

Abstract

This benchmark suite consists a number of examples of autonomous multi-agent systems where the agent number ranges from two to ten. The benchmarks are derived from the field of position-based formation control in autonomous robotics and vehicles. Their models are given as network of hybrid automata in the SpaceEx XML model format and can be transformed to other verification tools model formats using HyST, a model transformation tool. Safety of a small benchmark with two agents is analyzed using SpaceEx.

Category: academic **Difficulty:** low through challenge

1 Context and Origins

Intelligent autonomous systems have been a “hot” research topic for many years because of its rigorous application domains such as robotics, unmanned aerial vehicles (UAV), autonomous cars and sensors networks. The challenges in modeling, analysis, design and testing a such intelligent system have attracted researchers from different disciplines such as biology, computer, communication and control. In an early step, the intelligent behavior called “flocking behavior” of a group of animals such as bird, insect and fish has been investigated deeply over decades in the field of biology [1]. The behavior has been first modeled and simulated using computer in [2]. This work has inspired a new field of modeling, control and design for autonomous systems which is now considerably an important topic for the next generation of modern technology.

Consensus and formation controls are two fundamental problems in designing an autonomous system performing an intelligent behavior. Control scientists have proposed numerous protocols over last decades to drive the system to achieve some control objectives [3–9]. Generally, to perform a specific task, the agents need to exchange their information and cooperate with each other over communication channels. The communication topology of an autonomous system describes in detail how the information *flow* in the system. The communication topology can be *static*, i.e. does not change over times, or *dynamic*, i.e. may change over times. It can also be *directed*, i.e. information *flows* in one direction over a connection between two agents, or *undirected*, i.e. the information *flows* in both directions over a connection between two agents. The communication topology expresses the sensing and communicating capacities of the agents which affect significantly to the stability, controllability and the convergence of an autonomous system. Graph theory has been proved as an powerful tool for modeling the communication topology and analyzing the controllability of autonomous systems [10].

Formation control for autonomous systems [7–9] is seeking control laws to guarantee that the agents move to pre-determined positions while keeping the system formation in some specific shapes when moving. Depending on the sensing and communicating capacities of the agents, i.e. the communication topology, the formation control strategies can be categorized into position-based, displacement-based and distance-based approaches [11]. One essential safety requirement for the system is that there is no collision when the agents are moving. The formation control strategies have shown *informally* the ability of the agents avoiding collision via simulation-based testing. To guarantee the safety of the system, its formal model need to be given and verified using formal verification techniques.

Toward safety and liveness requirements of autonomous systems, some control algorithms have been proposed and verified using formal verification techniques recently [12, 13]. In this context, the formal model of an autonomous system is given based on discrete time intervals and to guarantee the safety of the system, the controller usually can perform some particular actions to resolve the potential risks coming. The whole system is modeled as a labeled transition system and the safety and liveness requirements are written in form of linear temporal logic (LTL).

Inspired by above interesting works, in this paper, we obtain a set of autonomous systems benchmarks written in SpaceEx XML format. Each agent is modeled separately as a hybrid automaton and the whole system is a network of hybrid automata which is basically a composition of all agents. Different from [12, 13], these benchmarks have *continuous dynamics*. Therefore, their safety requirements can be verified using existing verification tools that support verifying continuous dynamics [14–17]. In addition, when the number of agents increases, the benchmark models become larger that makes them harder to be verified. Thus, the large benchmarks are useful for testing the scalability of verification tools.

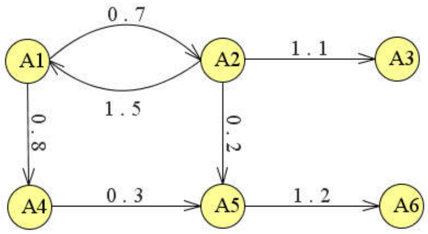
The rest of the paper is organized as follows: Section 2 presents the description of an autonomous system including the communication topology, the motion dynamics of the agents and the position-based formation control strategy. Section 3 presents different modeling approaches for an autonomous system. Section 4 analyzes a small benchmark with two agents using SpaceEx. Section 5 discusses some interesting issues for the future work and concludes the paper.

2 System descriptions

2.1 Communication topology

Directed/undirected graphs is powerful tool for modeling the interaction between agents in an autonomous system. In this benchmark suit, the communication topologies of all autonomous systems are modeled using directed graphs. A digraph (directed graph) defined by a tuple $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite non-empty set of vertices and $\mathcal{E} \in \mathcal{V}^2$ is a set of *ordered* pairs of vertices, called edges. It can be understood that vertice $v_i \in V$ represents for the i^{th} agent an autonomous system and ordered edge (i, j) represents for the interaction between the agent i and the agent j where the information flows from i to j, i.e. agent j receives the information from agent i. To model how much information flows in communication, we use a weighted digraph which can be defined by an adjacency matrix $A = [a_{ij}]_{n \times n}$, where $a_{ii} = 0$, $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ and $n = |\mathcal{V}|$ is the number of agents in the system. Figure 2.1 illustrates an example of communication topology of an autonomous system with six agents [18]. From the communication topology, it can be seen that one agent only can collect some information from its neighbors, not from all other agents.

A communication topology can be static, as in the case of the example, or dynamic, i.e. the connections between agents can be varied over times. A dynamic communication topology may be convenient to characterize naturally the interaction behaviors of agents in practice where the sensing capacity of agents is limited in some ranges and hence, it can not recognize the other agents outside of its sensing range. However, the dynamic communication topology increases the difficulty in designing the control law to guarantee autonomous systems to perform the intelligent flocking behavior. In this paper, the benchmarks can be categorized into static or



$$A = \begin{bmatrix} 0 & 1.5 & 0 & 0 & 0 & 0 \\ 0.7 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 & 0 \\ 0.8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.2 & 0 \end{bmatrix}$$

Figure 2.1: An example of communication topology using a weighted digraph.

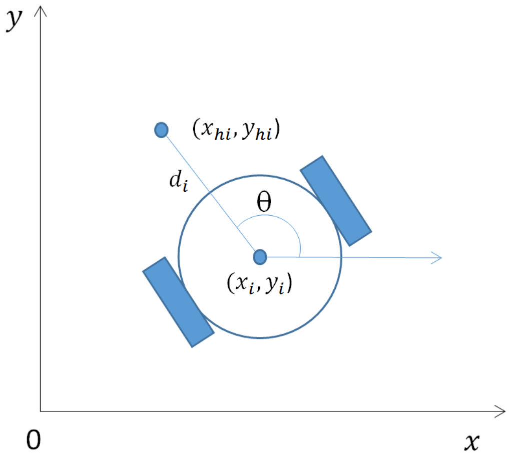


Figure 2.2: Nonholonomic differential driven mobile robot.

dynamic communication topology.

We have briefly introduced modeling interaction between agents in an autonomous system using directed graph. Next, we give the dynamics of the agents and the formation control rules of the autonomous system.

2.2 Motion dynamic and formation control

In this paper, we consider the formation control for multiple mobile robots in a 2-dimensional plan in which the equations of motion of a non-holonomic mobile robot depicted in Figure 2.2

are given by

$$\begin{aligned}
\dot{x}_i &= v_i \cos(\theta_i),\\
\dot{y}_i &= v_i \sin(\theta_i),\\
\dot{\theta}_i &= \omega_i,\\
m_i \dot{v}_i &= f_i,\\
J_i \dot{\omega}_i &= \tau_i,
\end{aligned} \tag{2.1}$$

where (x_i, y_i) is the Cartesian position of the robot centre, θ_i is the orientation, v_i is the linear velocity, ω_i is the angular velocity, m_i is the mass, J_i is the mass moment of inertia, f_i is the force, and τ_i is the torque applied to the robot.

Since Equation 2.1 contains the nonlinear functions $\cos(\theta_i)$ and $\sin(\theta_i)$. The robot dynamic is nonlinear and thus, we cannot model and analyze the system using SpaceEx. Fortunately, we can avoid the non-holonomic constraint and obtain a linear model for the system by introducing intermediate position variables (x_{hi}, y_{hi}) as follows [18].

$$\begin{bmatrix} x_{hi} \\ y_{hi} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + d_i \begin{bmatrix} \cos(\theta_i) \\ \sin(\theta_i) \end{bmatrix} \tag{2.2}$$

We can see that (x_{hi}, y_{hi}) is a position off the wheel axis of the i^{th} robot by a distance d_i. Now, if we let

$$\begin{bmatrix} f_i \\ \tau_i \end{bmatrix} = \begin{bmatrix} \frac{1}{m_i}\cos(\theta_i) & -\frac{d_i}{J_i}\sin(\theta_i) \\ \frac{1}{m_i}\sin(\theta_i) & -\frac{d_i}{J_i}\cos(\theta_i) \end{bmatrix}^{-1} \begin{bmatrix} v_{xi} + v_i\omega_i \sin(\theta_i) + d_i\omega_i^2\cos(\theta_i) \\ v_{yi} - v_i\omega_i\cos(\theta_i) + d_i\omega_i^2\sin(\theta_i) \end{bmatrix}$$

Then we can obtain the new linear equations of motion for each robot as a double-integrator system:

$$\begin{aligned}
\dot{x}_{hi} &= v_{xi},\\
\dot{v}_{xi} &= b_{xi},\\
\dot{y}_{hi} &= v_{yi},\\
\dot{v}_{yi} &= b_{yi}.
\end{aligned} \tag{2.3}$$

The control objective is to drive the mobile robots from their initial location (x_{hi}^0, y_{hi}^0) to predefined destinations (x_{hi}^d, y_{hi}^d) while preserve the formation of the system during the transition, e.g. a square formation for 4-robots team, a triangle formation for 3-robots team. Assume we have the communication topology of the system defined by adjacent matrix $A = [a_{ij}]_{n\times n}$, where n is the number of agents in the system, the formation control law for the system is designed as follows [18].

$$\begin{aligned}
b_{xi} &= -\alpha_x(x_{hi} - x_{hi}^d) - \gamma_x\alpha_x\dot{x}_{hi} - \sum_{j=1}^{n} a_{ij}[(x_{hi} - x_{hi}^d) - (x_{hj} - x_{hj}^d)] - \sum_{j=1}^{n}\gamma_x a_{ij}(\dot{x}_{hi} - \dot{x}_{hj})\\
b_{yi} &= -\alpha_y(y_{hi} - y_{hi}^d) - \gamma_y\alpha_y\dot{y}_{hi} - \sum_{j=1}^{n} a_{ij}[(y_{hi} - y_{hi}^d) - (y_{hj} - y_{hj}^d)] - \sum_{j=1}^{n}\gamma_y a_{ij}(\dot{y}_{hi} - \dot{y}_{hj})
\end{aligned} \tag{2.4}$$

where $\alpha_* > 0$ and $\gamma_* > 0$.

The first two terms of the control law are responsible for driving each robot to its destination (goal seeking) while the last two terms of the control law are to preserve the formation between robots (formation keeping). In term of verification, there are both liveness and safety properties need to be verified. The liveness property relates to goal seeking objective as we need to guarantee that each robot *finally reach its destination.* The safety property concerns the formation keeping problem as it is required there is *no collision* when robots are moving.

With above formation control law, we can derive the *closed-loop* dynamic equation for the system. Let $x_{ei} = x_{hi} - x^d_{hi}$, $y_{ei} = y_{hi} - y^d_{hi}$, $x_e = [x_{e1}, ..., x_{en}]^T$ and $y_e = [y_{e1}, ..., y_{en}]^T$, the closed-loop dynamic of the system can be written by

$$\begin{aligned} \begin{bmatrix} \dot{x}_e \\ \ddot{x}_e \end{bmatrix} &= \begin{bmatrix} 0_{n\times n} & I_n \\ -(L + \alpha_x I_n) & -\gamma_x(L + \alpha_x I_n) \end{bmatrix} \begin{bmatrix} x_e \\ \dot{x}_e \end{bmatrix} \\ \begin{bmatrix} \dot{y}_e \\ \ddot{y}_e \end{bmatrix} &= \begin{bmatrix} 0_{n\times n} & I_n \\ -(L + \alpha_y I_n) & -\gamma_y(L + \alpha_y I_n) \end{bmatrix} \begin{bmatrix} y_e \\ \dot{y}_e \end{bmatrix} \end{aligned} \tag{2.5}$$

where $0_{n\times n}$ is n-dimensional square zero matrix, I_n is n-dimensional identity matrix and $L = [l_{ij}]_{n\times n}$ in which $l_{ii} = \sum_{j\neq i} a_{ij}$ and $l_{ij} = -a_{ij}$, where $i \neq j$.

We have already described the communication topology, the system dynamics and formation control law. Next, we discuss how to model the system using hybrid automata.

3 System modeling

There are three approaches for modeling an autonomous system using hybrid automata framework. The first approach is that we can model the system using *decentralized style* in which each agent as a hybrid automata network composed by *dynamic component* describing the dynamic of the agent as defined in Equation 2.3 and *controller component* describing the distributed formation control law in Equation 2.4. Since the communication topology of the autonomous system may change, the controller component may switch its operation between different modes. The whole system will be a network of hybrid automata composing n agent's models. In the second approach, we can model the system using *centralized style* in which each agent is a single-mode hybrid automaton describing the dynamic of the agent, the control law given in Equation 2.4 is modeled as a *centralized coordinator* which is a hybrid automata containing one or multiple modes. Last but not least, we can also model the system as one single automaton describing the closed-loop dynamic defined by Equation 2.5.

The first two modeling approaches have two advantages. First, they describe intuitively the hierarchical architecture of the system in which each agent is a separate entity. The obtained model in the first modeling approach illustrates the *decentralized control strategy* in autonomous systems where the control signal is computed at agent side. In contrast to decentralized control strategy, the second approach describes the *centralized control strategy* where the coordinator collects the information of all agents and computes the control signals before sending them to the agents. Second, since the first two modeling approaches separate the agent's dynamic and the control law, they are convenient for changing the dynamics of the agents and they also allow modeling the switching happen between different dynamics of one agent. In addition, it is easy to model and verify the system under a complex hybrid control law when the controller switches between different modes along with communication topology changes. While the first two modeling approaches are convenient for modeling complex autonomous systems, the third approach is useful for finding an abstraction for the whole system that allows us to verify a very

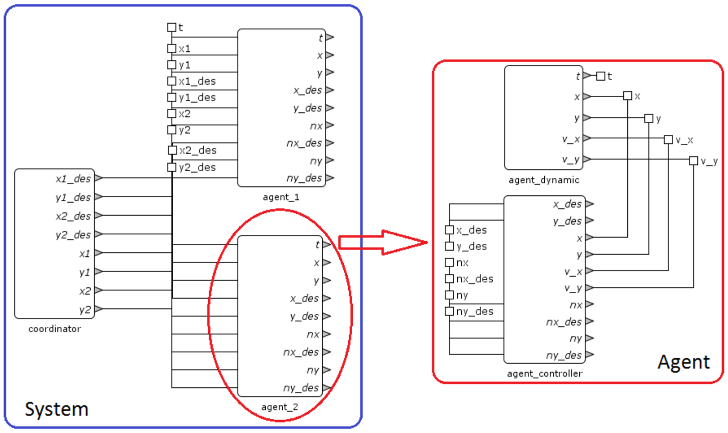


Figure 3.1: *Decentralized-style approach* for modeling autonomous systems using hybrid automata network.

large autonomous system using order-reduction abstraction method [**?**]. In this benchmark suit, we use the first and the second approaches to model distributed autonomous systems. Examples of these modeling approaches are depicted in Figure 3.1 and Figure 3.2.

4 Reachability analysis

The benchmark suit is presented in Table 4.1. In this paper, we present briefly the safety analysis of the benchmark $MAS2$ with two agents. The communication topology of $MAS2$ shows that the robot 2 receive the information from the robot 1. The initial *immediate positions* of two robots are (x^0_{h1}, y^0_{h1}) and (x^0_{h2}, y^0_{h2}) where $(0 \leq x^0_{h1} \leq 0.2, 0 \leq y^0_{h1} \leq 0.1)$ and $(0 \leq x^0_{h2} \leq 0.2, 0.9 \leq y^0_{h2} = 1)$. Assume that the distances between the intermediate positions and their corresponding robot centers are $d_1 = d_2 = l = 0.1$. The robots are controlled to go to their *immediate destinations* $(x^d_{h1} = 3, y^d_{h1} = 3)$ and $(x^d_{h2} = 4, y^d_{h2} = 4)$ while keeping their *intermediate distance* $d_h \geq 1$ as moving. The system is considerably safe if the *distance* between two robots, i.e. between the centers of two robots, is always larger than a threshold $d_{min} = 0.5$. The parameters for the distributed control law in Equation 2.4 are chosen as follows: $\alpha_x = 2\alpha_y = 2$, $\gamma_x = 2\gamma_y = 1$.

Figure 4.1 describes the trajectories of two robots. The figure shows that two robots finally reach their destinations.

Remind that (x_{h1}, y_{h1}) and (x_{h2}, y_{h2}) are not the centers of the robots as given in Equation 2.2. To verify the system safety, let $dis_x = x_2 - x_1$, $dis_y = y_2 - y_1$, $dis_{xh} = x_{h2} - x_{h1}$, $dis_{yh} = y_{h2} - y_{h1}$, the unsafe region of the system can be defined by the following circle.

$$|dis_x|^2 + |dis_y|^2 \leq d^2_{min}$$

If the unsafe region can not be reached, then two robots are always far away from each other at a distance $d > d_{min}$ and then, we can conclude that the system is safe. The reachable sets of dis_x and dis_y can be derived by *bloating* the reachable sets of dis_{xh} and dis_{yh} using the following constraints which are derived directly from Equation 2.2.

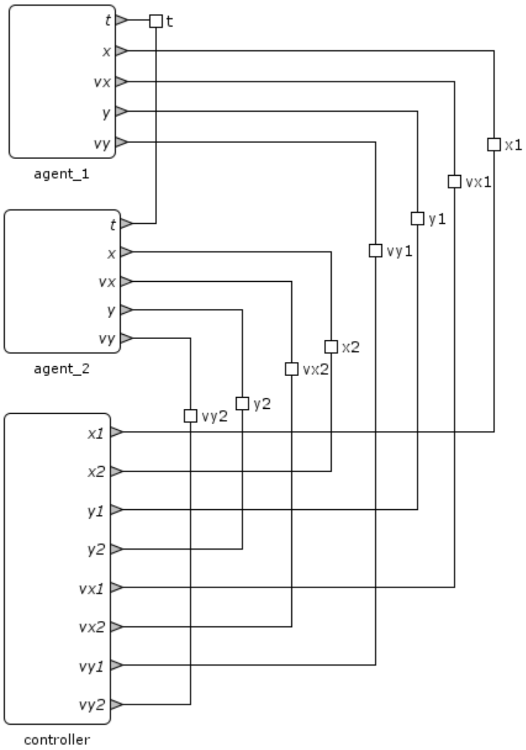


Figure 3.2: *Centralized-style approach* for modeling autonomous systems using hybrid automata network.

$$|x_2 - x_1| \leq |x_{h2} - x_{h1}| + 2l$$
$$|y_2 - y_1| \leq |y_{h2} - y_{h1}| + 2l$$

Figure 4.2 and Figure 4.3 illustrate the reachable set of dis_{xh} and dis_{yh} over times and Figure 4.4 describes the reachable set of (dis_x, dis_y) (the green polygon) which is bloated from the reachable set of (dis_{xh}, dis_{yh}) (the green polygon). The later figure shows that (dis_x, dis_y) does not reach the unsafe region for all times when the robots move to their destinations. Thus, we can conclude that the system is safe. In addition, we can see that the formation control law actually works since it drives the robots to their destinations and preserve the formation of the robots when they are moving (i.e. the intermediate distance between the robots finally converge to $d_h = \sqrt{2}$.

It is worth to notice that the control parameters $\{\alpha_x, \alpha_y, \gamma_x, \gamma_y\}$ assigned in Equation 2.4 affects significantly to the performance of the system. As analyzed in [18], there exists conditions for the control parameters and the communication topology to guarantee that the robots can finally reach their destination while preserving their formation.

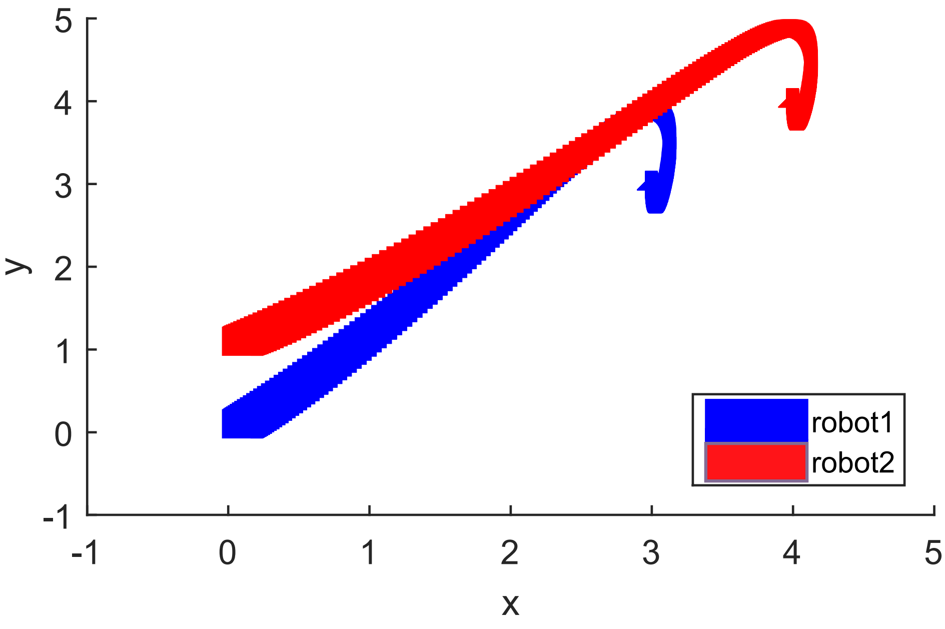


Figure 4.1: Trajectories of two robots.

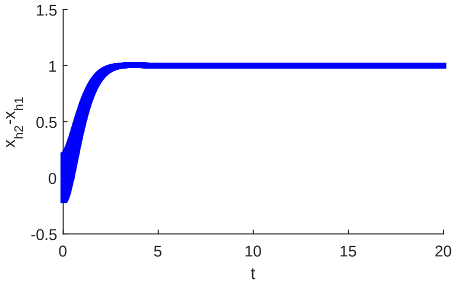


Figure 4.2: Reachable set of $dis_{xh} = x_{h2} - x_{h1}$ over times

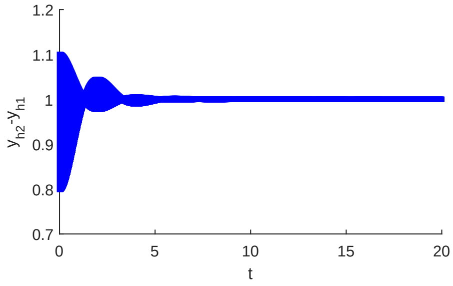


Figure 4.3: Reachable set of $dis_{yh} = y_{h2} - y_{h1}$ over times

5 Outlook

Overall, we present in this paper a set benchmarks for distributed autonomous systems, modeled as network of hybrid automata in the SpaceEx model format. The number of the agents range from two to ten. The position-based formation control has been successfully verified in a

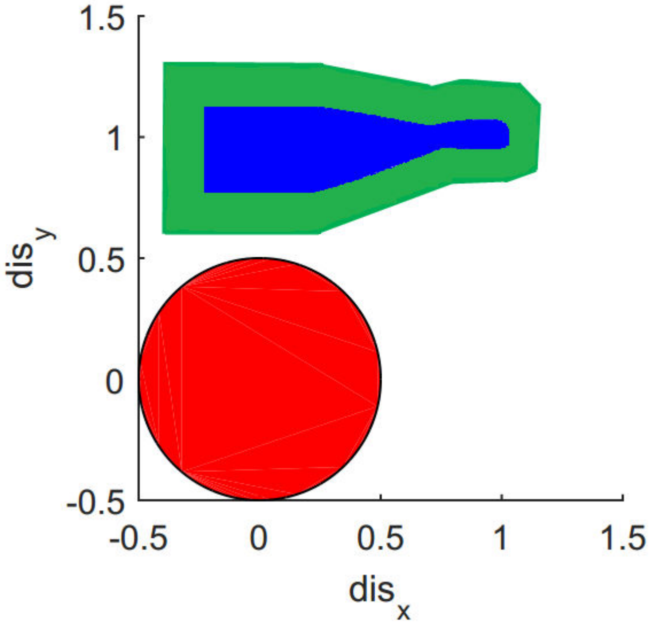


Figure 4.4: Reachable set of (dis_x, dis_y) (the green polygon), (dis_{xh}, dis_{yh}) (the green polygon) and the unsafe region (inside the red circle).

benchmark with two agents. There are two important issues should be considered in future work. First, it is challenging to model and verify the safety and livenesss properties of the distributed autonomous systems controlled by complex nonlinear formation control laws to avoid collision and obstacles. We can take advantages of verification tools supporting nonlinear hybrid systems such as Flow* [17] and C2E2 [19] in this case. Second, it would be useful for testing verification tools if we can generate automatically distributed autonomous systems with arbitrary large number of agents. We are going to implement this feature as an automatic generator in Hyst [20].

Acknowledgements The material presented in this paper is based upon work supported by the National Science Foundation (NSF) under grant numbers CNS 1464311 and CCF 1527398, the Air Force Research Laboratory (AFRL) through contract number FA8750-15-1-0105, and the Air Force Office of Scientific Research (AFOSR) under contract number FA9550-15-1-0258. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of AFRL, AFOSR, or NSF.

References

[1] H. R. Pulliam, “On the advantages of flocking,” *Journal of theoretical Biology*, vol. 38, no. 2, pp. 419–422, 1973.

[2] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 25–34, 1987.

[3] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[4] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *IEEE Transactions on automatic control*, vol. 51, no. 3, pp. 401–420, 2006.

[5] T. Balch and R. C. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE transactions on robotics and automation*, vol. 14, no. 6, pp. 926–939, 1998.

[6] W. Ren and R. W. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on automatic control*, vol. 50, no. 5, pp. 655–661, 2005.

[7] F. Xiao, L. Wang, J. Chen, and Y. Gao, “Finite-time formation control for multi-agent systems,” *Automatica*, vol. 45, no. 11, pp. 2605–2611, 2009.

[8] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE transactions on robotics and automation*, vol. 17, no. 6, pp. 947–951, 2001.

[9] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, “Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments,” *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[10] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt, “Controllability of multi-agent systems from a graph-theoretic perspective,” *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 162–186, 2009.

[11] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” *Automatica*, vol. 53, pp. 424–440, 2015.

[12] T. Johnson and S. Mitra, “Safe flocking in spite of actuator faults,” in *Symposium on Self-Stabilizing Systems.* Springer, 2010, pp. 588–602.

[13] L. Bobadilla, T. T. Johnson, and A. LaViers, “Verified planar formation control algorithms by composition of primitives,” in *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 1541.

[14] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, “Spaceex: Scalable verification of hybrid systems,” in *Computer Aided Verification.* Springer, 2011, pp. 379–395.

[15] S. Kong, S. Gao, W. Chen, and E. Clarke, “dreach: δ-reachability analysis for hybrid systems,” pp. 200–205, 2015.

[16] S. Bak, “Hycreate: A tool for overapproximating reachability of hybrid automata,” *Retrieved January*, vol. 17, p. 2016, 2013.

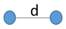
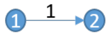
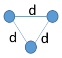
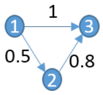
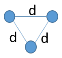
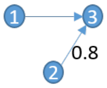
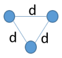
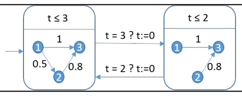
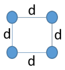
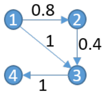
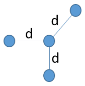
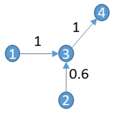
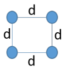
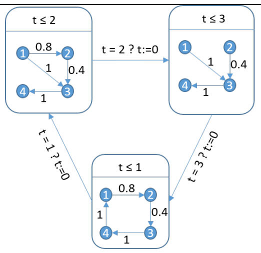
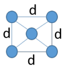
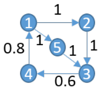
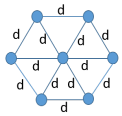
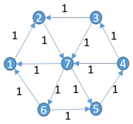
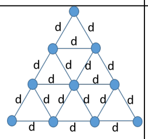
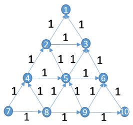
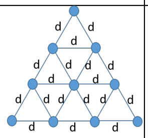
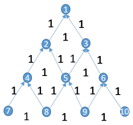
[17] X. Chen, E. Ábrahám, and S. Sankaranarayanan, “Flow*: An analyzer for non-linear hybrid systems,” in *International Conference on Computer Aided Verification.* Springer, 2013, pp. 258–263.

[18] W. Ren and E. Atkins, “Distributed multi-vehicle coordinated control via local information exchange,” *International Journal of Robust and Nonlinear Control*, vol. 17, no. 10-11, pp. 1002–1033, 2007.

[19] P. S. Duggirala, S. Mitra, M. Viswanathan, and M. Potok, “C2e2: a verification tool for stateflow models,” in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems.* Springer, 2015, pp. 68–82.

[20] S. Bak, S. Bogomolov, and T. T. Johnson, “Hyst: a source transformation and translation tool for hybrid automaton models,” in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control.* ACM, 2015, pp. 128–133.

Table 4.1: Benchmark collection

No.	Benchmarks	n	Formation	Communication Topology
1	$MAS2$	2		
2	$MAS3_1$	3		
3	$MAS3_2$	3		
4	$MAS3_3$	3		
5	$MAS4_1$	4		
6	$MAS4_2$	4		
7	$MAS4_3$	4		
8	$MAS5$	5		
9	$MAS7$	7		
10	$MAS10_1$	10		
11	$MAS10_2$	10		
12	$MAS10_3$	10