The 2016 AAAI Fall Symposium Series:
Cross-Disciplinary Challenges for Autonomous Systems
Technical Report FS-16-03

Extended Abstract: Formal Design of Cooperative Multi-Agent Systems*

Rafael Rodrigues da Silva†, Bo Wu, Jin Dai, Hai Lin
Department of Electrical Engineering
University of Notre Dame, Notre Dame, IN, 46556 USA
rrodri17@nd.edu, bwu3@nd.edu, jdai1@nd.edu, hlin1@nd.edu

Background

Cooperative multi-agent systems refer to a class of multi-agent systems in which a number of possibly heterogeneous agents collaborate autonomously through spatially-distributed physical interactions and wireless communication networks. Compared with a monolithic system, cooperative multi-agent systems offer greater efficiency and flexibility due to the redundancy of functionalities, ability to re-configure and robustness to uncertain environments, and therefore show great potential in a wide variety of applications, ranging from power grids, transportation systems, computer networks to robotic teams, see e.g. (Arkin 1998; Choset et al. 2005; Fainekos et al. 2009b; Lin 2014) and the references therein.

A key issue in cooperative multi-agent systems is how to design local control policies for each agent as well as coordination strategies among them such that certain desirable specifications can be satisfied. Existing methods can roughly be divided into two categories: *bottom-up* and *top-down* approaches. In the bottom-up design, local interactions and control policies are pre-defined with inspirations from natural or social behaviors (Zavlanos et al. 2009; Arkin 1998), and non-trivial global behaviors emerge from these local controllers and their interactions. Representative examples include behaviour based robotics (Arkin 1998) and consensus based approaches (Olfati-Saber and Murray 2004). In contrast, the top-down design relies on a "divide-and-conquer" coordination and control scheme, and decomposes a global mission into local task specifications (Karimadini and Lin 2011) or distributes a global cost function into local utility functions (Marden, Shamma, and others 2012) for each agent based on their individual sensing and actuating capabilities. The bottom-up approach scales well but generally lacks formal performance guarantees, except for certain properties like consensus (Olfati-Saber and Murray 2004), rendezvous (Dimarogonas and Kyriakopoulos 2007) or related formation control (Fax and Murray 2004). Top-down design, on the other hand, can provide performance guarantees, but lacks flexibility and scalability in local control policy design. For example, in symbolic motion planning the space is normally partitioned into many labeled regions and no moving objects other than the agents are assumed (Belta et al. 2007b). Therefore, it is difficult to handle dynamic environments. Additionally, the planning complexity quickly becomes prohibitively high as the number of partitioned regions and agents increase, which further hampers the applicability of the abstraction based methods in many practical circumstances.

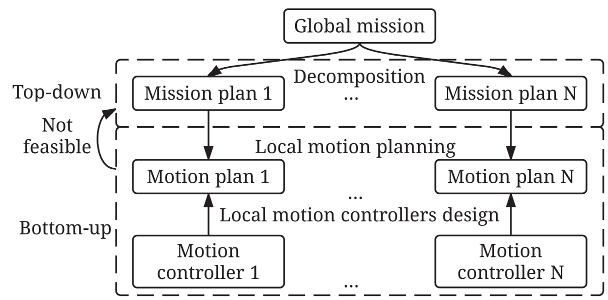

Figure 1: Overall framework

Our Idea

Hence, we are motivated to combine top-down and bottom-up design methods so to leverage both advantages and pave the way towards a scalable, adaptive and automatic design method for cooperative robotic teams with performance guarantees. For such a purpose, cross-disciplinary approaches combining methods from control theory, machine learning, and computational verification are pursued in our study. More specifically, we propose a formal design framework for cooperative multi-agent systems by combining top-down mission planning with bottom-up motion-planning. In this work, we assume that the multi-agent system is assigned a global mission, specified as regular languages over all the agents' capabilities, whereas basic motion controllers for each agent shall be designed with respect to a given environment description. We propose to use a layered control architecture, as illustrated in Fig. 3, for each intelligent agent.

From the top, a mission planning layer decomposes the global mission into local tasks that are consistent with

*This work is supported by NSF-CNS-1239222, NSF-EECS-1253488 and NSF-CNS-1446288

†The first author would like to appreciate the scholarship support by CAPES/BR, BEX 13242/13-0

each agent's individual capabilities and compositionally verifies the joint effort of the agents via an assume-guarantee paradigm. On the other hand, corresponding to these local missions, motion plans associated with each agent are synthesized by composing basic motion primitives, which are verified safe by differential dynamic logic (d$\mathcal{L}$), through a Satisfiability Modulo Theories (SMT) solver that searches feasible solutions in the face of constraints due to local task requirements and the environment description. It is shown that the proposed framework can handle changing environments as the motion primitives are reactive in nature, making the motion planning adaptive to local environmental changes. Furthermore, on-line mission reconfiguration can be triggered by the motion planning layer once no feasible solutions can be found through the SMT solver. Our use of SMT is inspired by recent work on SMT-based robot motion planning (Hung et al. 2014; Saha et al. 2014; Nedunuri et al. 2014). Instead of using SMT for motion planning only, we also encode high-level mission specifications into SMT and solve for an integrated mission and motion plan.

The next two sections will give more details on the top-down mission decomposition and bottom-up motion composition with a team of mobile robots as a working example.

Cooperative Mission Decomposition of Cooperative Multi-agent Systems

We assume that the global specification for the cooperative multi-agent system is given as a prefix-closed regular language K^{MI} over the global mission set Σ_{MI}, where $\Sigma_{MI} = \bigcup_{i \in \mathcal{N}_\mathcal{A}} \Sigma^i_{MI}$ and Σ^i_{MI} represents the motion primitives for the i-th agent. Intuitively speaking, Σ^i_{MI} contains those local motion/actions that the i-th agent can achieve, where symbols in both Σ^i_{MI} and Σ^j_{MI} require collaboration between the agent i and j.

Given K^{MI} and Σ^i_{MI} for $i \in \mathcal{N}_\mathcal{A}$, we pursue a top-down design methodology and aim to decompose the global task into feasible local tasks, based on each agent's sensing and actuating capabilities. A counterexample-guided and learning-based formal synthesis framework was proposed in our previous work (Dai and Lin 2014), which is shown in the top-down layer in Fig. 3 and is illustrated in Fig. 2 in more detail to automatically learn the local missions K^i_{MI}.

The automatic coordination framework executes the following stages iteratively to achieve cooperative task decomposition and mission planning among the agents.

- *Initial task allocation* A prefix-closed and feasible local mission K^i_{MI} for agent i from the global mission K_{MI} is initially obtained by $K^i_{MI} = P_i(K_{MI})$, where P_i stands for the *natural projection* (Cassandras and Lafortune 2008b) from the global mission set Σ_{MI} to the mission set Σ^i_{MI} of the i-th agent.
- *Compositional verification* We determine whether or not the collective behaviors of each agent can satisfy the global mission by deploying a compositional verification (Dai and Lin 2014) procedure with each behavior module being a component DFA that recognizes K^i_{MI}. In particular, to mitigate the computational complexity, we adopt an assume-guarantee paradigm (Partovi and Lin 2014) for the compositional verification and modify the L^* algorithm (Angluin 1987) to automatically learn appropriate assumptions for each agent.
- *Counterexample-guided synthesis* If the local missions fail to satisfy the global specification jointly, the compositional verification returns a counterexample indicating that all the $K^i_{MI}, i \in \mathcal{N}_\mathcal{A}$ share a same illegal trace that violates the global mission. We present such counterexamples to re-synthesize the local missions.

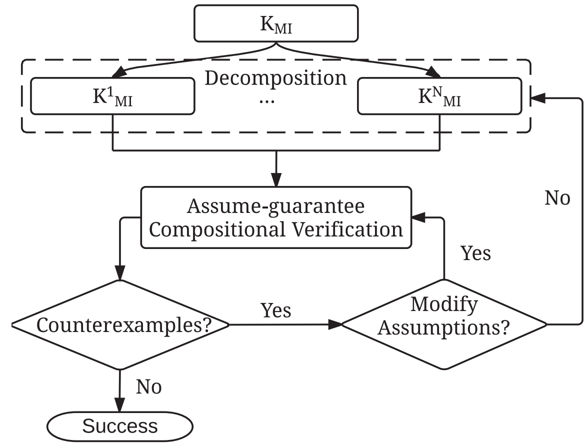

Figure 2: Learning-based coordination and mission planning framework.

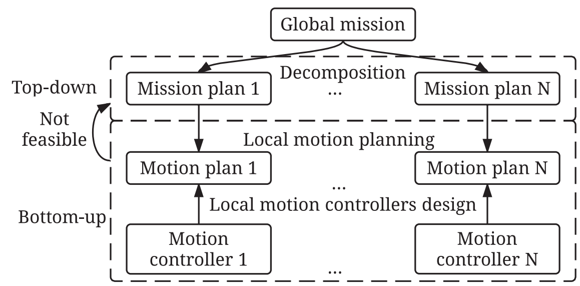

Figure 3: Overall framework

Bottom-up Design and Motion Planning

In the Bottom-up layer, we adopt a hierarchical planner inspired by behavior based robotics (Nakhaeinia and Karasfi 2013) distributed in each robot and consisting of two layers: global and local. The global layer synthesizes an integrated task and motion plan composing certificated reactive controllers, which we call safe motion primitives, that satisfies a local task specification for the respective robot. The local layer implements the global plan executing those primitives as planned. Hence, the performance is guaranteed through bottom-up modular verifications. First, safe motion primitives are designed with verified performances. Then, a global plan is built upon these certified controllers. Since the method proposed here is of a bottom-up and compositional nature, we call it as CoSMoP (Composition of Safe Motion Primitives).

To design safe motion primitives, we propose to formally verify the reactive controllers that we call safe motion primitives in Differential Dynamic Logic (d$\mathcal{L}$) (Platzer 2010), for which verification software tools are available, e.g., KeYmaera (Platzer 2010). For example, we can use the Dynamic Window Approach (DWA) (Fox et al. 1997) as a primitive for ground vehicles to drive a robot while avoiding unexpected obstacles that can be even moving. DWA algorithms are a widely adopted and efficient approach for mobile robots to avoid collisions in uncertain and dynamic environments. The safety of a DWA algorithm that extends collision avoidance for moving obstacles has been formally proved in (Mitsch, Ghorbal, and Platzer 2013) using d$\mathcal{L}$ and hybrid system verification. This proof allows us to abstract this primitive to the global layer, where the task and motion plans are integrated.

To synthesize a global plan, those safe motion primitives are abstracted as Counter Linear Temporal Logic over Constraint System CLTLB($\mathcal{D}$) (Bersani et al. 2010) formulas, which we call the motion primitives specification, that is encoded with the local task specification to a Satisfiability Modulo Theories (SMT) solver. This plan comprises pairs of actions (i.e. safe motion primitives) and waypoints (i.e. goal states), which is the sequential execution of actions that the robot must perform to ensure a task specification formally. The CoSMoP encodes an integrated task and motion problem to SMT by extending Bounded Satisfiability Checking (BSC) (Pradella, Morzenti, and Pietro 2013). The BSC models consist of temporal logic rather than transition systems; thus, the problem encoding is more compact and elegant. Moreover, when using the Counter Linear Temporal Logic over Constraint System CLTLB($\mathcal{D}$) (Bersani et al. 2010) language, it was also shown that if the constraint system $\mathcal{D}$ is decidable, then so is the CLTLB($\mathcal{D}$) formula, and it can be encoded to SMT (Bersani et al. 2010). Therefore, encoding the integrated task and motion problem using CLTLB($\mathcal{D}$) language allows the description of a wide range of system properties in a problem that is decidable. Besides, using those primitives, we formally ensure that the robot will always be in a safe state. Thus, if the environment is fair, meaning that any moving obstacles will always eventually leave their actual state or an unexpected static obstacle does not lead any robot to a deadlock, then the plan will satisfy the local task specification. If not, since the robots are always safe, we can update the environment description adding new obstacles using the sensors readings, for example, and search for new plans at current state in a receding horizon strategy.

Related Work and our Contributions

Our study is influenced by recent advances in robot *symbolic motion planning*, which uses formal methods to generate a symbolic path on an abstracted quotient system to satisfy temporal logic specifications, see e.g., (Belta et al. 2007a; Fainekos et al. 2009a; Kress-Gazit, Fainekos, and Pappas 2009; Kloetzer and Belta 2010; Wongpiromsarn, Topcu, and Murray 2010; Chen et al. 2012). The high-level missions are usually specified as temporal logic formulas (Pnueli 1977) due to their expressiveness and similarity to natural languages (Finucane, Jing, and Kress-Gazit 2010). The basic design procedure for symbolic motion planning consists of the following steps: First, a finite abstracted model of the robotic system is obtained. Then, the design is carried out in the discrete domain using methods like model checking (Clarke, Grumberg, and Peled 1999; Baier, Katoen, and Larsen 2008), reactive synthesis (Pnueli and Rosner 1989; Piterman, Pnueli, and Saar 2006) and supervisory control theory (Ramadge and Wonham 1987; Cassandras and Lafortune 2008a) to generate feasible runs consisting of sequences of discrete (symbolic) states that satisfy the temporal logic specifications of concern. Finally, the generated sequence of discrete states or symbols is used by the continuous layer to construct continuous feedback control laws to drive the robot (or physically feasible trajectories for robots to follow). The critical step, also the most difficult part, of symbolic motion planning is how to obtain an abstraction of the robotic dynamics and environment. The abstracted finite model should be constructed in such a way that, once one can find a run in the abstracted model satisfying the specification, there must exist corresponding continuous trajectories for the original robotic system satisfying the same specification. Most of research efforts in the literature have been devoted to answering the abstraction problem, using bisimulation (Alur et al. 2000; Tabuada and Pappas 2003; 2006) and approximate bisimulation based abstraction (Girard and Pappas 2009; Tabuada 2009), maneuver automata (Frazzoli, Dahleh, and Feron 2005), and multi-affine control induced workspace partitions (Belta, Isler, and Pappas 2005; Kloetzer and Belta 2008). To avoid abstraction, there are some recent efforts to encode Linear Temporal Logic or Signal Temporal Logic specifications into mixed-integer constraints and then use optimization based approaches to find feasible solutions (Raman et al. 2014). However, the bottleneck of this method lies in its computational complexity, as the mixed-integer constraint solver does not scale well as the dimension of the problem increases. To mitigate the computational complexity issue, receding horizon planning (Wongpiromsarn, Topcu, and Murray 2012) techniques were usually adopted. However, this restricts the method to be effective only for bounded-time temporal formulas, as it is generally inconclusive to just check a finite prefix of a trajectory for unbounded-time properties. Furthermore, the physical dynamics that can be handled is still limited as general non-linear constraints are difficult to handle in the mixed-integer solver. In addition, the uncertainties handled so far were restricted to bounded external disturbances in the physical dynamics, while a more interesting robustness/resilience property of intelligent agents is how they adapt in unstructured and uncertain dynamic environments through learning.

In the context of literature, our main contributions lie in

- Our introduction of a new formal framework to solve both the mission and the motion planning problems of cooperative multi-agent systems, based on which provably correct mission plans and motion controllers are designed.

- Our proposed design framework shows good potential in scalability. On one hand, in the top-down mission planning stage, we use an assume-guarantee paradigm (Păsăreanu et al. 2008) to compositionally verify the correctness of all the mission plans; on the other hand, we synthesize the corresponding motion controllers by using an SMT solver and thus fine partitioning of the environment is avoided.
- Our proposed design framework provides reactive solutions for both mission and motion planning problems. First, we develop a modification of the L^* learning algorithm (Angluin 1987) such that it can be applied for local mission planning even if the agent's model is not known *a priori*; secondly, by composing simple motion primitives, our designed motion controllers show great reactiveness to uncertain environments.

References

Alur, R.; Henzinger, T.; Lafferriere, G.; and Pappas, G. 2000. Discrete abstractions of hybrid systems. *Proceedings of the IEEE* 88(7):971 –984.

Angluin, D. 1987. Learning regular sets from queries and counterexamples. *Information and computation* 75(2):87–106.

Arkin, R. C. 1998. *Behavior-based robotics*. MIT press.

Baier, C.; Katoen, J.-P.; and Larsen, K. G. 2008. *Principles of model checking*. MIT press.

Belta, C.; Bicchi, A.; Egerstedt, M.; Frazzoli, E.; Klavins, E.; and Pappas, G. 2007a. Symbolic planning and control of robot motion. *Robotics and Automation Magazine, IEEE* 14(1):61–70.

Belta, C.; Bicchi, A.; Egerstedt, M.; Frazzoli, E.; Klavins, E.; and Pappas, G. J. 2007b. Symbolic planning and control of robot motion [grand challenges of robotics]. *IEEE Robotics & Automation Magazine* 14(1):61–70.

Belta, C.; Isler, V.; and Pappas, G. 2005. Discrete abstractions for robot motion planning and control in polygonal environments. *Robotics, IEEE Transactions on* 21(5):864 – 874.

Bersani, M. M.; Frigeri, A.; Morzenti, A.; Pradella, M.; Rossi, M.; and Pietro, P. S. 2010. Bounded reachability for temporal logic over constraint systems. In *Temporal Representation and Reasoning (TIME), 2010 17th International Symposium on*, 43–50. IEEE.

Cassandras, C. G., and Lafortune, S. 2008a. *Introduction to Discrete Event Systems (2nd Edition)*. Springer-Verlag.

Cassandras, C. G., and Lafortune, S. 2008b. *Introduction to discrete event systems*. Springer Science & Business Media.

Chen, Y.; Ding, X.; Stefanescu, A.; and Belta, C. 2012. Formal approach to the deployment of distributed robotic teams. *Robotics, IEEE Transactions on* 28(1):158 –171.

Choset, H.; Lynch, K.; Hutchinson, S.; Kantor, G.; Burgard, W.; Kavraki, L.; and Thrun, S. 2005. *Principles of robot motion: theory, algorithms, and implementations*. MITPress, Boston.

Clarke, E. M.; Grumberg, O.; and Peled, D. 1999. *Model Checking*. MIT press.

Dai, J., and Lin, H. 2014. Automatic synthesis of cooperative multi-agent systems. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 6173–6178.

Dimarogonas, D. V., and Kyriakopoulos, K. J. 2007. On the rendezvous problem for multiple nonholonomic agents. *IEEE Transactions on automatic control* 52(5):916–922.

Fainekos, G. E.; Girard, A.; Kress-Gazit, H.; and Pappas, G. J. 2009a. Temporal logic motion planning for dynamic robots. *Automatica* 45(2):343–352.

Fainekos, G. E.; Girard, A.; Kress-Gazit, H.; and Pappas, G. J. 2009b. Temporal logic motion planning for dynamic robots. *Automatica* 45(2):343–352.

Fax, J. A., and Murray, R. M. 2004. Information flow and cooperative control of vehicle formations. *IEEE transactions on automatic control* 49(9):1465–1476.

Finucane, C.; Jing, G.; and Kress-Gazit, H. 2010. LTL-MoP: Experimenting with language, temporal logic and robot control. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 1988–1993. IEEE.

Fox, D.; Burgard, W.; Thrun, S.; et al. 1997. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine* 4(1):23–33.

Frazzoli, E.; Dahleh, M. A.; and Feron, E. 2005. Maneuver-based motion planning for nonlinear systems with symmetries. *IEEE Trans. on Robotics* 21(6):1077C1091.

Girard, A., and Pappas, G. J. 2009. Hierarchical control system design using approximate simulation. *Automatica* 45(2):566–571.

Hung, W. N.; Song, X.; Tan, J.; Li, X.; Zhang, J.; Wang, R.; and Gao, P. 2014. Motion planning with satisfiability modulo theories. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 113–118. IEEE.

Karimadini, M., and Lin, H. 2011. Guaranteed global performance through local coordinations. *Automatica* 47(5):890–898.

Kloetzer, M., and Belta, C. 2008. A fully automated framework for control of linear systems from temporal logic specifications. *Automatic Control, IEEE Transactions on* 53(1):287–297.

Kloetzer, M., and Belta, C. 2010. Automatic deployment of distributed teams of robots from temporal logic motion specifications. *Robotics, IEEE Transactions on* 26(1):48 – 61.

Kress-Gazit, H.; Fainekos, G.; and Pappas, G. 2009. Temporal-logic-based reactive mission and motion planning. *Robotics, IEEE Transactions on* 25(6):1370 –1381.

Lin, H. 2014. Mission accomplished: An introduction to formal methods in mobile robot motion planning and control. *Unmanned Systems* 2(02):201–216.

Marden, J. R.; Shamma, J. S.; et al. 2012. Game theory and distributed control. *Handbook of game theory* 4:861–900.

Mitsch, S.; Ghorbal, K.; and Platzer, A. 2013. On provably safe obstacle avoidance for autonomous robotic ground vehicles. In *Robotics: Science and Systems*.

Nakhaeinia, D., and Karasfi, B. 2013. A behavior-based approach for collision avoidance of mobile robots in unknown and dynamic environments. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology* 24(2):299–311.

Nedunuri, S.; Prabhu, S.; Moll, M.; Chaudhuri, S.; and Kavraki, L. E. 2014. Smt-based synthesis of integrated task and motion plans from plan outlines. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 655–662. IEEE.

Olfati-Saber, R., and Murray, R. M. 2004. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control* 49(9):1520–1533.

Partovi, A., and Lin, H. 2014. Assume-guarantee cooperative satisfaction of multi-agent systems. In *2014 American Control Conference*, 2053–2058. IEEE.

Păsăreanu, C. S.; Giannakopoulou, D.; Bobaru, M. G.; Cobleigh, J. M.; and Barringer, H. 2008. Learning to divide and conquer: applying the l* algorithm to automate assume-guarantee reasoning. *Formal Methods in System Design* 32(3):175–205.

Piterman, N.; Pnueli, A.; and Saar, Y. 2006. Synthesis of reactive (1) designs. In *Verification, Model Checking, and Abstract Interpretation*, 364–380. Springer.

Platzer, A. 2010. *Logical analysis of hybrid systems: proving theorems for complex dynamics*. Springer Science & Business Media.

Pnueli, A., and Rosner, R. 1989. On the synthesis of a reactive module. In *Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 179–190. ACM.

Pnueli, A. 1977. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, 46–57. IEEE.

Pradella, M.; Morzenti, A.; and Pietro, P. S. 2013. Bounded satisfiability checking of metric temporal logic specifications. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 22(3):20.

Ramadge, P. J., and Wonham, W. M. 1987. Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization* 25(1):206–230.

Raman, V.; Donzé, A.; Maasoumy, M.; Murray, R. M.; Sangiovanni-Vincentelli, A.; and Seshia, S. A. 2014. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, 81–87. IEEE.

Saha, I.; Ramaithitima, R.; Kumar, V.; Pappas, G. J.; and Seshia, S. A. 2014. Automated composition of motion primitives for multi-robot systems from safe LTL specifications. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, 1525–1532. IEEE.

Tabuada, P., and Pappas, G. 2003. From discrete specifications to hybrid control. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 4, 3366–3371 vol.4.

Tabuada, P., and Pappas, G. 2006. Linear time logic control of discrete-time linear systems. *Automatic Control, IEEE Transactions on* 51(12):1862–1877.

Tabuada, P. 2009. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer New York: Springer.

Wongpiromsarn, T.; Topcu, U.; and Murray, R. 2010. Receding horizon control for temporal logic specifications. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, HSCC '10, 101–110.

Wongpiromsarn, T.; Topcu, U.; and Murray, R. M. 2012. Receding horizon temporal logic planning. *IEEE Transactions on Automatic Control* 57(11):2817–2830.

Zavlanos, M. M.; Tanner, H. G.; Jadbabaie, A.; and Pappas, G. J. 2009. Hybrid control for connectivity preserving flocking. *IEEE Transactions on Automatic Control* 54(12):2869–2875.