Disjunctive Fuzzy Control: Application to Swarms

Robert J. Marks II
Ph.D., IEEE Fellow, OSA Fellow, ARSG
Distinguished Professor of Electrical & Computer Engineering
Department of Electrical & Computer Engineering
Baylor University, Waco, Texas

Abstract

Given simple agent rules, a swarm's emergent behavior can be difficult to predict. The inverse problem is typically even more difficult: given a desired emergent behavior, what are the rules by which swarm agents should obey? Disjunctive fuzzy control is proposed as a method to model swarm agents. Compared to more commonly used conjunctive fuzzy control such as proposed by Mamdami, disjunctive fuzzy control is robust and disjointed connected. Instead of agents working together to achieve a goal, each swarm agent contributes individually to the result. As is the case with social insect swarms, the emergent behavior displays gracious degradation as agents are lost. Disjunctive fuzzy control allows adaptation of the describing membership function, as is commonly done in conjunctive control.

1 Introduction

Swarm intelligence [5] is based on the emergent behavior of individual social insect agents performing simple tasks. Swarm intelligence has found application is telecommunications [10], [13], business [6], robotics [4], [16], and optimization [11, 12], makes use of a plurality of highly disjoint agents interacting using simple rules. Simple swarm algorithms have been employed to assist with load balancing of peer-to-peer networks [21], routing within mobile ad hoc radio networks [15], and self-organizing construction and assembly [17].

Often, determination of emergent behavior from simple rules of interaction in swarm intelligence escapes both analytic and intuitive inspection [14].

1. Agents roam a wood particle covered floor picking up particles in their paths. Each agent roams around randomly until it bump into a second particle wherein they unload their load. Now empty handed, they continue roaming looking for another particle and the process is repeated.[5]
2. In a large group of agents, each agent randomly identifies two other agents and moves to place themselves between two agents [6].
3. In a large group of agents, each agent, say X, identifies at random two other agents we label Y and Z. The agent X moves place agent Y between itself and Z [6].

These are three examples of simple rules that can characterize a swarm of hundreds or thousands of agents. The rules are expressed clearly and without ambiguity. The identification of the emergent behavior of the swarm, however, is not readily evident in these three cases.[1]

This three simple examples illustrate the difficulty of the analysis of emergent behavior in even simple swarms. The inversion of the swarm, or swarm design, is more daunting. Given a desired emergent behavior, what are the set of simple rules needed? We investigate such a design applied to predator-prey swarms.

2 Swarm Formulation and Control

Disjunctive fuzzy logic [7, 8, 25, 26, 27, 28] is ideally suited to the control of swarms.

[1] Once an emergent behavior is identified, however, the relationship between the rules and the emergent behavior can become more clear. The emergent behavior for these three cases is in the Appendix (Section 3.

2.1 Logic Statement

Let $\mathcal{C}$ be the consequent emergent behavior of a collection of swarm agents, $\{\mathbb{A}_n | n \leq n \leq N\}$. An attempt to control the performance of the swarm using traditional conjunctive implication can be expressed

$$\bigcap_{n=1}^{N} \mathbb{A}_n \to \mathbb{C}. \tag{1}$$

Disjunctive implication, more conducive to swarm behavior, is

$$\bigcup_{n=1}^{N} (\mathbb{A}_n \to \mathbb{C}) \tag{2}$$

For the Boolean case, there is an identity between the two implication types

$$\left[\bigcap_{n=1}^{N} \mathbb{A}_n \to \mathbb{C}.\right] \equiv \left[\bigcup_{n=1}^{N} (\mathbb{A} \to \mathbb{C})\right]. \tag{3}$$

For fuzzy logic, there is not an identity. There is, however, a similarity of performance which can often be made equivalent [27].

In fuzzy form, (2) describes traditional fuzzy Mamdami control whose use ranges from automobile transmissions to kitchen appliances [19]. The conjunctive form in (1) can be written as

$$\texttt{If (} \mathbb{A}_1 \texttt{ AND } \mathbb{A}_2 \texttt{ AND ... AND } \mathbb{A}_N \texttt{), then } \mathbb{C}. \tag{4}$$

The disjunctive equivalent in (2) is

$$\begin{gathered}\texttt{(If } \mathbb{A}_1 \texttt{ then } \mathbb{C}\texttt{) OR (If } \mathbb{A}_2 \texttt{ then } \mathbb{C}\texttt{) OR} \\ \texttt{... OR (If } \mathbb{A}_N \texttt{ then } \mathbb{C}\texttt{)}\end{gathered} \tag{5}$$

The disjunctive form of implication is seen to be conducive to description of disjoint agents contributing to a common consequent as the case with social swarms. Like a swarm, agents can be lost or added and the goal of the collective remains the same. The agents can be identical yet collectively contribute to the swarm's goal. The conjunctive form in (4), by comparison, is brittle. Loss of an agent requires reassessment of the implication structure. Details contrasting the characteristics of conjunctive and disjunctive implications can be found elsewhere.[7, 8, 25, 26, 27, 28]

2.2 Agent

Each of the N agents, $\{\mathbb{A}_n\}$, has its own control. For swarm environments, we can assume all of the agents are identical.[2] A model of an single agent is shown in Figure 1. Sensors on the nth agent, $\mathbb{A}_m$, generate the control antecedents, $\{\mathcal{A}_m | 1 \leq m \leq M\}$. There are K consequences, $\{\mathcal{C}_k | 1 \leq \ell \leq K\}$ for the agent. The consequences are aggregated as an instruction to the agent actuator, $\mathcal{P}$. We will assume in our model there is a single agent actuator.[3]

When the control process of each agent is fixed, the emergent behavior is determined at least in a probabilistic sense. Emergence can vary depending, for example, on agent initialization, environmental variations, and stochastic performance components such as jitter. The degree to which the swarm is resilient to such variations can be determined by applying appropriate cost to the swarm performance, $\mathbb{C}$, which, for a desired performance attribute can be viewed as a random variable.

[2] In other words, the swarm is homogenous and has a single goal. The agents therefore do not switch among operating modes. In some models of homogeneous swarms, there are stochastic variations among agents that are important, for example, in switching smoothly among various modes of operation,[5] *e.g.* worker ants converting to soldier ants when their colony is being attacked.

[3] See the caption of Figure 1 for a more detailed example.

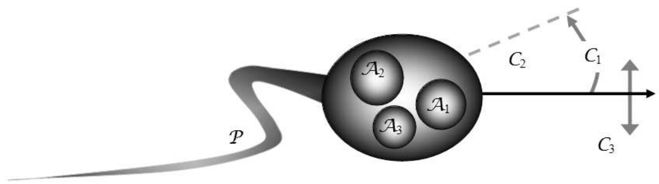

Figure 1: Illustration of control at the agent level. In this example, there are three sensors generating antecedents. The consequents are $\mathcal{C}_1$ = linear acceleration, $\mathcal{C}_2$ = angular acceleration, and $\mathcal{C}_3$ = jitter. (Jitter is a random perturbation useful in swarming and predator evasion.) The three consequents must be aggregated into a single motion control action, $\mathcal{P}$, which will be constrained by agent parameters such as maximum speed and physical parameters such as momentum.

2.2.1 Disjunctive Agent Control

Here are some reasons to choose disjunctive rather than conjunctive control over swarm agents.

- **Adaptability.** In a four wheel vehicle, one can turn left by

  $\mathcal{A}_1$: turning the front wheels to the left,

  $\mathcal{A}_2$: turning the rear wheels to the left,

  $\mathcal{A}_3$: braking the wheels on the left side, or

  $\mathcal{A}_4$: accelerating the wheels on the right side more than the left.

  Conjunctive control, for $N = 4$ is

  $$\bigcap_{n=1}^{N} \mathcal{A}_n \to \mathcal{C}$$

  achieves the consequent goal, $\mathcal{C}$, of turning left by assigning contributions of each of the antecedents, $\{\mathcal{A}_n\}$. In essence, if the steering wheel figures in the control (antecedents $\mathcal{A}_1$ and $\mathcal{A}_2$) and we loose the steering wheel, we have to back to the drawing board to reconfigure the "turn left" rules.

  Disjunctive control allows each antecedent to contribute to the consequent,

  $$\bigcup_{n=1}^{N} (\mathcal{A}_n \to \mathcal{C}) \tag{6}$$

  If steering is lost, we can still turn left by braking. No reconfiguration is required since no interaction or assignment of relative antecedent contribution has been made in the [4]

- **Linear Versus Exponential Growth in Search Space.** If there are N antecedents and a single consequent, each requiring $\{F_n | 1 \leq n \leq N\}$ fuzzy sets, the Mamdami fuzzy rule matrix for implementation requires

  $$R_\cap = \prod_{n=1}^{N} F_n$$

  entries. If all of the $F = F_n$'s are the same, then $R_\cap = F^N$. Disjunctive control, on the other hand, requires

  $$R_\cup = \sum_{n=1}^{N} F_n$$

[4] Amusingly, the simple "swarm rule" for NASCAR agents can be stated as "Go fast - turn left." [20].

rules. If the number of fuzzy sets is the same, then $R_\cup = NF$. The number of fuzzy rules therefore increases linearly with respect to the number of antecedents rather than exponentially. In the inversion of the swarm, we will be searching through a space whose dimension is determined by a the number of rules. Therefore, besides its operational advantages, disjunctive control reduces the search space size thereby avoiding the *curse of dimensionality* [23] for the inversion process.

- **Functional representation.** The nth term in the disjunctive expression in (6) is $\mathcal{A}_n \to \mathcal{C}$. Assume the antecedent $\mathcal{A}_n$ is tesselated into F_n fuzzy membership functions. As is with the case of traditional Mamdami fuzzy control, the sensor reading is fuzzified into a vector of F_n membership values. Since there is a single antecedent, these values are used to weight the fuzzy membership functions of the consequent which are then defuzzified into a single crisp consequent, c_n. From this process, we conclude that each sensor reading is assigned a single consequent value. Thus, *the value of the consequent is a direct function of the antecedent* which we can write as

$$c_n = \zeta_n(a_n) \tag{7}$$

  where a_n is the reading of the nth sensor. We refer to the $\zeta_n(\cdot)$'s as *actuator functions.* The structure of the actuator functions can be viewed as being formed from the structure of the fuzzifying and defuzzifying membership functions.

  Although use of the fuzzification and defuzzification membership functions has been the past practice in disjunctive fuzzy control [7, 8, 25, 26, 27, 28], the determination of the actuator functions can be performed directly without consideration of these intermediate components. Each of the four rules for turning left, for example, are increasing functions of antecedents. Although not themselves fuzzy membership functions, the actuator functions can generally be chosen with the same heuristic freedom as fuzzy membership functions.

  The consequent values, $\{c_n\}$, are then combined into a single consequent, C. From (6), a disjunctive aggregation such as a t-conorm is appropriate. Let the disjunctive aggregation be denoted by the multidimensional function.

$$C = \theta\left(c_1, c_2, \cdots, c_N\right). \tag{8}$$

  In summary, disjunctive control consists of the following steps.

  1. Acquire the N antecedent values $\{a_n\}$,
  2. Evaluate the corresponding consequents $c_n = \zeta(a_n)$ using (7).
  3. Conjunctively aggregate the $\{c_n\}$'s into a single value of using C using (8).

  As is done with fuzzy membership functions in conjunctive Mamdami type control, the actuator function can be parameterized and the parameters adapted [1, 2, 3, 18]

- **Agent consequent as the swarm antecedent.** As is illustrated in Figure 2, the agent consequent, c, becomes the swarm antecedent. There will typically be more than one agent consequent. The opacity of the explanation facility for this interface manifests itself in the difficulty of relating individual agent actions to the swarm's emergent behavior. The inversion required for swarm design is likewise intuitively illusive.

- **Generalization and summary.** If we allow M consequents per agent, then (8) can be written

$$C[m] = \theta\left(c_1[m], c_2[m], \cdots, c_N[m]\right); 1 \le m \le M \tag{9}$$

  where (7) is now

$$c_n[m] = \zeta_{nm}(a_n) \tag{10}$$

  The flow of control is therefore

$$\begin{aligned} \{a_n\} \quad &\Longrightarrow \quad c_n[m] = \zeta_{nm}(a_n) \\ &\Longrightarrow \quad C[m] = \theta\left(c_1[m], c_2[m], \cdots, c_N[m]\right) \\ &\equiv \quad \mathbb{A} \\ &\Longrightarrow \quad \mathbb{C}. \end{aligned} \tag{11}$$

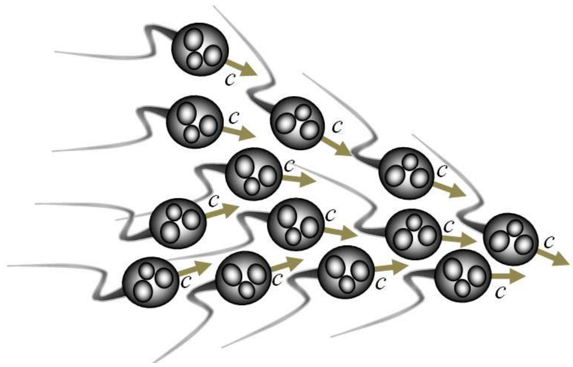

Figure 2: The agent consequent, c, becomes the swarm antecedent. There will typically be more than one agent consequent.

The single swarm consequent, $\mathbb{C}$, is therefore determined by this for directly from the agent antecedents.[5] Performance in the forward implication chain is dictated by environmental and physical constraints. For a fixed set of agent antecedents, $\{a_n\}$, the swarm consequent is determined by the actuator functions, $\zeta_{nm}(\cdot)$, and the *disjunctive aggregation functions*, $\theta(\cdot)$. These functions can be parameterized and, to effect inversion, the parameters adapted to yield a desired swarm performance, $\mathbb{C}$. In a search process, this can be done by assigning a fitness metric to the consequent and searching for agent antecedents that result in maximization.

3 Appendix

The emergent behavior resulting from the three simple swarm rules in the Introduction (Section 1) are, as follows,

1. This is a simple model used for termites stacking wood or ants collecting their dead into piles.
2. This is the *protector* mode analyzed by Gravagne and Marks [14]. The emergent behavior is attraction of all the agents to a fixed set of points. If, in the coupling, all agents can be linked in the rules to all other agents, convergence is to a single point.
3. This is the *refugee mode* mode analyzed by Gravagne and Marks [14], all agents diffusively disperse.

References

[1] P. Arabshahi, R.J. Marks II and T.P. Caudell, "Adaptation of Fuzzy Inferencing: A Survey", Proceedings of the IEEE/Nagoya University WWW on Learning and Adaptive Systems, pp.1-9, October 22-23, 1993, Nagoya University, (Nagoya, Japan)

[2] P. Arabshahi, J.J. Choi, R.J. Marks II and T.P. Caudell, "Fuzzy Parameter Adaptation in Optimization: Some Neural Net Training Examples", Computational Science and Engineering, (IEEE Computer Society), vol 3, No 1, Spring 1996, pp.57-65.

[3] P. Arabshahi, R.J. Marks II, S. Oh, T.P. Caudell , J.J. Choi, and B.G. Song, "Pointer Adaptation and Pruning of Min-Max Fuzzy Estimation," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol.44, no.9, September 1997, pp.696-709.

[5] Multiple objectives in swarm inversion have not been studied.

[4] R. Beckers, O. E. Holland, and J. L. Deneubourg, “From local actions to global tasks: Stigmergy and collective robotics,” in Prog. Artificial Life IV. Cambridge, MA: MIT Press, 1994, pp. 181189.

[5] E Bonabeau, M. Dorigo, and G. Thereulaz, *Swarm Intelligence: From Natural to Artificial Systems*, NY: Oxford Univ. Press, 1999

[6] E. Bonabeau and C. Meyer, “Swarm intelligence, a whole new way to think about business,” Harvard Bus. Rev., vol. 79, no. 5, pp. 106114, May 2001

[7] W. E. Combs and J. E. Andrews, “Combinatorial rule explosion eliminated by a fuzzy rule configuration,” IEEE Trans. Fuzzy Systems, vol. 6, no. 1, pp. 1-11, Feb. 1998.

[8] William E. Combs, Jeffrey J. Weinschenk, Robert J. Marks II, “Genomic Systems Design: A novel, biologically-based framework for enhancing the adaptive, autonomous capabilities of computer systems”, FUZZ-IEEE 2004, IEEE International Conference on Fuzzy Systems, 25-29 July, 2004, Budapest.

[9] William A. Dembski and Robert J. Marks II, “Conservation of Information in Search: Measuring the Cost of Success,” IEEE Transactions on Systems, Man and Cybernetics A, Systems & Humans, vol.5, #5, September 2009, pp.1051-1061

[10] G. Di Caro and M. Dorigo, “AntNet: Distributed stigmergetic control for communications networks,” J. Artif. Intell. Res., vol. 9, pp. 317365, 1998.

[11] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 26, no. 1, pp. 2941, Feb. 1996.

[12] M. Dorigo and T. Stützle, *Ant Colony Optimization.* Cambridge, MA: MIT Press, 2004.

[13] I. Kassabalidis, M. A. El-Sharkawi, R. J. Marks, P. Asabshahi, and A. A. Gray, “Swarm intelligence for routing in communication networks,” in Proc. IEEE Globecom, San Antonio, TX, Nov. 2001, pp. 36133617.

[14] I.A. Gravagne and R.J. Marks II, ”Emergent Behaviors of Protector, Refugee and Aggressor Swarm,” IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, Volume 37, Issue 2, April 2007, pp. 471 - 476.

[15] M. Günes, U. Sorges, and I. Bouazizi, “ARAThe ant-colony based routing algorithm for MANETs,” in Proc. IEEE ICPPW, 2002, pp. 7985.

[16] A. T. Hayes, A. Martinoli, and R. M. Goodman, “Swarm robotic odor localization,” in Proc. IEEE/RSJ Int. Conf. IROS, Oct. 2001, pp. 10731087.

[17] H. Leung, R. Kothari, and A. Minai, “Phase transition in a swarm algorithm for self-organizing construction,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 68, no. 4, pp. 046 111.1046 111.5, 2003.

[18] R.J. Marks II, S.Oh, P. Arabshahi, T.P. Caudell, J.J. Choi and B.G. Song, ”Steepest descent of min-max fuzzy if-then rules”, Proceedings of the International Joint Conference on Neural Networks, Beijing, vol. III, pp. 471-477, November 3-6, 1992.

[19] R.J. Marks II, Editor, *Fuzzy Logic Technology and Applications,* IEEE Technical Activities Board, Piscataway, 1994.

[20] Joseph McRae Mellichamp, *Go Fast, Turn Left*, ISBN #978-1427602008

[21] A. Montresor, H. Meling, and Ö. Babaoglu, “Messor: Load-balancing through a swarm of autonomous agents,” Dept. Comput. Sci., Univ. Bologna, Bologna, Italy, Tech. Rep. UBLCS-2002-11, 2002.

[22] Russell D. Reed, R.J. Marks II and Seho Oh, “Similarities of error regularization, sigmoid gain scaling, target smoothing and training with jitter”, IEEE Transactions on Neural Networks, vol. 6, no.3, May 1995, pp. 529-538.

[23] Russell D. Reed and R.J. Marks II, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*, (MIT Press, Cambridge, MA, 1999.)

[24] Benjamin B. Thompson, Robert J. Marks II, Mohamed A. El-Sharkawi, Warren J. Fox, and Robert T. Miyamoto, "Inversion of Neural Network Underwater Acoustic Model for Estimation of Bottom Parameters Using Modified Particle Swarm Optimizers", 2003 International Joint Conference on Neural Networks, July 20-24, 2003, Portland, Oregon (pp. 1301-1306).

[25] Jeffrey J. Weinschenk, William E. Combs, Robert J. Marks II, "Avoidance of rule explosion by mapping fuzzy systems to a disjunctive rule configuration", 2003 International Conference on Fuzzy Systems (FUZZ-IEEE), St. Louis, May 25-28, 2003. systems", FUZZ-IEEE 2004, IEEE International Conference on Fuzzy Systems, 25-29 July, 2004, Budapest.

[26] Jeffrey J. Weinschenk, Robert J. Marks II, William E. Combs, "Layered URC fuzzy systems: a novel link between fuzzy systems and neural network," 2003 International Joint Conference on Neural Networks, July 20-24, 2003 , Portland , Oregon (pp. 2995-3000).

[27] Jeffrey J. Weinschenk, Robert J. Marks II, William E. Combs, "On the use of Fourier methods in URC fuzzy system design", FUZZ-IEEE 2004, Proceedings 2004 IEEE International Conference on Fuzzy Systems, Budapest, Volume 2, 25-29 July 2004, pp. 911 - 916.

[28] Jeffrey J. Weinschenk, William E. Combs, Robert J. Marks II, "On the avoidance of rule explosion in fuzzy inference engines," International Journal of Information Technology and Intelligent Computing, vol.1, #4 (2007).