

Technical report 08-002

# **A general modeling framework for swarms\***

J. van Ast, R. Babuška, and B. De Schutter

*If you want to cite this report, please use the following reference instead:*

J. van Ast, R. Babuška, and B. De Schutter, “A general modeling framework for swarms,” *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC 2008)*, Hong Kong, pp. 3796–3801, June 2008.

Delft Center for Systems and Control  
Delft University of Technology  
Mekelweg 2, 2628 CD Delft  
The Netherlands  
phone: +31-15-278.24.73 (secretary)  
URL: <https://www.dcsc.tudelft.nl>

---

\*This report can also be downloaded via [https://pub.deschutter.info/abs/08\\_002.html](https://pub.deschutter.info/abs/08_002.html)

# A General Modeling Framework for Swarms

Jelmer van Ast, Robert Babuška, Bart De Schutter  
Delft Center for Systems and Control  
Delft University of Technology  
Mekelweg 2, 2628 CD Delft, the Netherlands

**Abstract**—Swarms are characterized by the ability to generate complex behavior from the coupling of simple individuals. While the swarm approach to distributed systems of moving agents is gradually finding a way to engineering applications, a true successful demonstration of an engineered swarm is still missing. One of the reasons for this is the gap between the complexity of the swarms studied in fundamental research and the complexity needed for the application to interesting control problems. In the majority of the research on swarm intelligent systems, the moving agents in the swarm are modeled as simple reactive agents. This model comprises too little intelligence to fully exploit the potential of swarms. In this paper, a general comprehensive swarm framework is introduced and related to the established state of the art. Such a framework is novel and it is a first and important step in the development and analysis of more complex and intelligent swarms.

## I. INTRODUCTION

IN the recent years, research on moving cooperative autonomous agents has shown that their collective behavior can be controlled to solve particular optimization and control problems. In the majority of the publications, the moving agents are modeled as simple reactive agents characterized by a state denoting their position and a function describing their dynamics, and they are frequently called particles. The collection of these particles is called a *swarm* and its application is referred to as *swarm intelligence* [1]. The individual agents are typically too simple for the label ‘intelligent’, although the swarm often does manifest intelligent behavior. Swarms of particles have been successfully applied to optimization and control problems, where the control objective for the particles is to form a cohesive swarm in a certain formation. This paper proposes a general framework for swarms of moving agents, which combines the notions of particles and dynamic agents to facilitate the development of swarms of more intelligent agents.

Swarming is a term from biology denoting the collective motion and behavior of a group of insects, bacteria, or animals [2], [3]. Typically, swarms complete tasks that are impossible to complete by the individuals alone. A swarm consists of cooperating autonomous individuals, each having their own strategies, while they are not aware of the global objective. Each individual only locally interacts with the environment and communicates with its neighbors. There is neither a supervisor, nor a hierarchical structure. These characteristics

This research is financially supported by Senter, Ministry of Economic Affairs of The Netherlands within the BSIK-ICIS project “Self-Organizing Moving Agents” (grant no. BSIK03024). Bart De Schutter is also with the department of Marine and Transport Technology, Delft University of Technology.

provide some intrinsic advantages to the engineering application of swarms:

- Simple homogeneous individuals can be produced in series, resulting in lower production costs.
- The swarm is potentially robust to errors in the individuals. Individuals can be made of relatively unreliable components, further reducing production costs. Malfunctioning individuals can easily be replaced.
- The system is scalable; individuals can easily be added, or removed.

Some conceptual developments of swarm applications have been presented for platooning and the navigation of ground robots and unmanned aerial vehicles. Platooning is the formation of a string of vehicles that control their speed to maintain a desired inter-vehicle distance and is closely related to the subject of one-dimensional swarms. In such swarms, the moving agents are modeled as particles with proximity sensors at both sides. Mainly, the stability is studied under various conditions of the communication links [4], [5]. The control of multiple cooperative robotic vehicles controlling a formation, guarding a perimeter, and surrounding a facility is analyzed in [6]. Cooperative transport by robots, based on similar behavior by several species of ants is considered in [7]. In [8], a detailed description is given of the simulation and possible realization of a sensor Unmanned Aerial Vehicle (UAV), theoretically capable of autonomous refueling, cooperative search, information fusion, and munition deployment. The analysis of a group of micro-UAVs, capable of performing multiple target detection and localization is discussed in [9]. Swarms of sensors can facilitate detecting and discriminating targets with low signal-to-noise ratio by allowing correlation between different sensor types and/or different aspect angles.

Only a few applications of real swarm robotic systems have been presented. A swarm of micro robots that communicate at a short distance and have a perception of their surroundings by IR sensors is described in [10]. Another application is the *swarm-bot* project, which constitutes a self-assembling and self-organizing robotic system [11]. The mobile robots have the ability to connect to each other and to disconnect from each other, and the research focuses on the problem of synthesizing controllers for the swarm-bots using artificial evolution. Specifically, aggregation and coordinated motion of the swarm-bot are studied.

Although the swarm approach to distributed systems of moving agents is gradually finding a way to engineering applications, true successful demonstrations of engineered swarms

are still lacking. One of the reasons for this is a ‘missing link’ between the fundamental research on particle swarm behavior and the application to practically relevant control problems. This paper introduces a general swarm framework that supports such a link by combining particle swarms and dynamic agents. It must be stressed that a proper modeling framework has not yet been presented in the literature and that it is an essential first step before any analysis can be made of the behavior of more intelligent moving agents in a swarm.

The rest of this paper is structured as follows. In Section II, the proposed framework is introduced and discussed in detail. Section III discusses two examples of how the framework relates to the state of the art and Section IV indicates its opportunities for future research and development and concludes this paper.

## II. SWARM FRAMEWORK

A swarm is a multi-agent system of cooperating autonomous agents. Generally, in multi-agent systems, the coordination of the agents is achieved by complex strategies, in a fixed topology. Often a central controller is used to determine the optimal action for each of the agents. Such methods, however, scale poorly with the number of agents. Swarm intelligence aims at controlling a large number of cooperative autonomous agents, in a varying topology, with simple, local rules. The analysis of a swarm intelligence system typically focuses on the dynamics of the swarm as a whole, rather than on the dynamics of the individual agents. The most prevalent control problem considered in swarm literature is that of swarm aggregation, which is briefly described in Section III-B. The results for swarm aggregation are too limited to be generally applicable to a wide class of control problems. This demonstrates the difficulty of applying swarm intelligence to practically relevant control problems. However, considering the many potential advantages of swarms, further research is highly desired.

This section presents a new framework that enables a more structured approach to developing swarm intelligence for distributed sensing and control and provides better insight in the structure of swarm systems. The framework separates the physical parts and behavior of the swarm members from their decision making capabilities. Current swarm intelligence research focuses mainly on the physical behavior of the swarm members, while more sophisticated decision making is studied in dynamic agent systems. The proposed framework aims to integrate both fields to enable the development and analysis of more sophisticated swarm systems.

In the considered framework, the members of the swarm are called *moving agents*. The two key features of a moving agent are that:

- 1) it can move through its environment and
- 2) it is capable of decision making based on its input and recollection of the past.

These features are represented by two strictly distinct classes, called *particles* and *dynamic agents*. A diagram of the framework is given in Fig. 1.

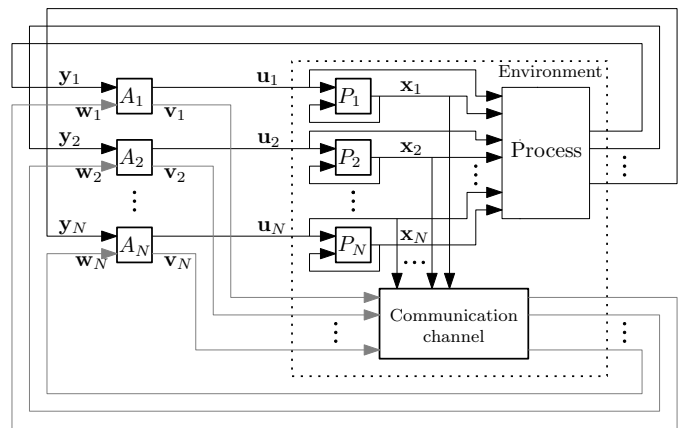


Fig. 1. Block diagram of the framework.

The diagram shows  $N$  dynamic agents  $A_1, \dots, A_N$  interacting with an environment. The dynamic agents represent the intelligence of the moving agents. The particles  $P_1, \dots, P_N$  represent the physical part of the moving agents, such as their position and speed. Each dynamic agent, indexed by  $i$ , senses the environment by a vector  $y_i$  and produces an input  $u_i$  to the environment. Furthermore, each dynamic agent may send and receive messages  $v_i$  and  $w_i$  to and from the environment, respectively. Within the communication channel, it is determined which messages are received by which dynamic agents, based on the state of all the particles  $\mathbf{x}_s = [\mathbf{x}_1^T \dots \mathbf{x}_N^T]^T$ . Each dynamic agent is associated with a particle by their corresponding index. The state of each particle changes based on the input from its associated dynamic agent and the restrictions posed on it by the environment. The process state may be sensed and influenced by the inputs  $\mathbf{u}_s = [\mathbf{u}_1^T \dots \mathbf{u}_N^T]^T$  from the dynamic agents and the state  $\mathbf{x}_s$  of all the particles. The rest of this section discusses the elements of the framework in more detail.

*Example 1:* In order to exemplify the discussion throughout this section, a swarm of identical tiny robots is considered, the task of which is to guard a building. This is clearly a task that may benefit from the advantages of a swarm over a single agent. As it is assumed that the robots are frequently cut off from long-range radio communication, a certain level of autonomy is required. It is also desired for such a system to be scalable and robust to malfunctioning robots. The robots are assumed to be able to drive over the terrain surrounding the building and they have some short-range communication capabilities, some processing power, and sensors for sensing some properties of their immediate environment.  $\square$

### A. Particles

In research on swarm intelligence, the members of the swarm are usually modeled as particles, though the exact definition of a particle is not always given. In this paper, a particle is defined as follows:

*Definition 2.1:* A particle  $P_i$  is an entity having a state  $\mathbf{x}_i \in X$ , containing the physical states of the moving agent.

Particles at least represent the position of the moving agent in the environment, but may include many other states, ranging from velocity and orientation to shape and color. The particle is in fact the placeholder of the moving agent and it is not responsible for the decision making. The states of the particles influence the neighborhood for all the agents, the local value of the sensor input, and the distance between the agents.

*Example 2:* The physical position and speed of each robot in our previous example belong to the state of its associated particle. The state changes as a result of the actions chosen by the robots, according to the dynamics of the robots and the environment. The particle may also represent other physical properties of the robots, such as size, shape, color, etc, depending on what is useful for the control problem.  $\square$

### B. Dynamic Agents

The decision making power of a moving agent is taken care of by a dynamic agent. Dynamic agents are the basis of modern artificial intelligence [12]. A dynamic agent is defined as an entity that observes the environment, possesses an internal state that changes as a function of the observations and acts on the environment based on this internal state and the observations. Moreover, it is able to send and receive messages, which is discussed further in Section II-F.

*Definition 2.2:* A dynamic agent  $A_i$ , is a tuple  $\langle Z, Y, U, W, h, \pi, \phi \rangle$ , with:

- $\mathbf{z}_i \in Z$  the agent's internal state,
- $\mathbf{y}_i \in Y$  its observation of the environment,
- $\mathbf{u}_i \in U$  its input to the environment,
- $\mathbf{w}_i, \mathbf{v}_i \in W$  its incoming and outgoing message respectively,
- $h : Z \times Y \times W \rightarrow Z$  its internal state transition function, describing how the internal state evolves as a result of observations and the messages received,
- $\pi : Z \times Y \times U \rightarrow [0, 1]$  its decision probability distribution, and
- $\phi : Z \rightarrow W$  its message generating function.

In this definition, the behavior of a dynamic agent is thus defined as a probabilistic mapping of states and observations to actions. This allows for explorative decisions, which is essential to decision making in an environment of which there is no model available to the dynamic agents. This stochasticity is explicitly separated from the dynamic agent's internal state transition function  $h$  to stress the algorithmic nature of  $h$ .

*Example 3:* The dynamic agent of each robot is the software routine that receives input from sensors and communication links, processes this input based on its current state, and produces an output. This output generally drives the actuators of the robot and usually results in a change of the position, thus a change of the particle state. According to this state and the environment, the dynamic agent is presented with other information based on which it has to make a new decision.

According to the principle that the physical state of the robot is not a part of the dynamic agent, the position of the robot is not known to the dynamic agent right away. Everything the dynamic agent needs to know, it has to draw from its observations of the environment.

The framework also stipulates that the decision making part does not necessarily needs to be present at the same location as the rest of the robot. Remote controlled robots are in this sense identical to robots with an onboard controller.  $\square$

For the application to swarms, dynamic agents are assumed to be *cooperative*. Cooperativeness is usually defined in the way that all the agents aim at achieving the same objective. Within the swarm framework, this needs a little more explanation, as the agents are all autonomous and thus act based on their own local objective function, or strategy. The control objective is generally defined for the whole swarm, so the cooperativeness must hold for a global objective that is unknown to the agents. As this is a *contradictio in terminis*, dynamic agents in a swarm are usually said to be cooperative if the agents do not aim at preventing other agents from achieving their individual goals.<sup>1</sup>

### C. Moving Agents

Using Definitions 2.1 and 2.2 a *moving agent* can be defined.

*Definition 2.3:* A moving agent  $M_i = (A_i, P_i)$  is a pair of a dynamic agent and a particle.

In this way, the dynamic agent can operate without directly taking into account its motion through the environment, and the motion of the particle can be considered without directly taking into account the decision making of the dynamic agent. Existing results on either part can be more easily combined to form more sophisticated swarms than the current state of the art. When discussing the behavior of the swarm, one can now refer to its members by the clearly defined moving agents. Regarding stability and performance of the swarm, conditions can be determined for the signals  $\mathbf{u}$  and  $\mathbf{y}$  and the messages  $\mathbf{w}$  and  $\mathbf{v}$  that couple the dynamic agents and the environment.

*Example 4:* The physical robots (particles) including their software for the decision making (dynamic agents) together form the moving agents.  $\square$

### D. Process

Everything that must be sensed, or controlled by the moving agent is called the process. It may include a real process, for instance a chemical reaction, but also a virtual process, such as an artificial potential field (see Section III-B).

*Definition 2.4:* A process is characterized by its state  $\psi \in \Psi$ , which contains the variables that must be sensed and/or controlled by the dynamic agents.

The process state may change based on the output of the dynamic agents  $\mathbf{u}_s$  and the state of the particles  $\mathbf{x}_s$ . Typically,

<sup>1</sup>A better term would be *indifferent* rather than cooperative.



the state of the process is distributed in space when a swarm approach is chosen to sense or control it. The observations of the dynamic agent may vary for varying positions as a result of this.

*Example 5:* Threats in the environment must be noticed by the robots, and their state forms the process. There may also be a virtual process, when the interaction between the robots and the environment is modeled by an artificial potential field.  $\square$

### E. Environment

As illustrated in Fig. 1, the environment encompasses everything that is outside of the dynamic agent. It consists of everything that is physically present in the problem setting. It holds the physical state of the moving agents  $\mathbf{x}_s$  (the particles), the process, and the communication channel. It also defines the state space of the particles, i.e., the world state with boundaries and obstacles. As the particles move through the environment and potentially are obstacles to each other, this state space is dynamic. The environment also includes the state transition functions of the particles.

The term environment, as used in this definition, comes from the dynamic agent community. It is different from what is standard in the systems and control community, where the environment denotes everything that is outside of the controller and the process, and is usually just held responsible for the disturbances of the signals in the system. Disturbances can also easily be added to the framework, as well as communication delays and errors, although these are left out from the discussion in this paper.

*Definition 2.5:* The *environment* is a tuple  $\langle N, X, g, \Psi, f, C \rangle$ , where

- $N$  is the number of particles,
- $X$  is the state space of the particles,
- $g : X \times \mathbf{U} \times X \rightarrow [0, 1]$ , with  $\mathbf{U} = \times_i^N U_i$  the joint action space, is the particle's state transition function, describing how the state of the particle evolves as a result of the action taken by all dynamic agents,
- $\Psi$  is the state space of the process,
- $f : \Psi \times \mathbf{U} \times \Psi \rightarrow [0, 1]$ , with  $\mathbf{U} = \times_i^N U_i$  the joint action space, is the process transition probability distribution, describing how the process evolves as a result of the agents' actions, and
- $C$  is the communication channel.

*Example 6:* The world outside to the software of the robots is the environment. It contains the physical robots, the building, its surroundings, possible threats, the communication signals, and other signals that may be received by the sensors of the robots.  $\square$

### F. Communication and Interaction

A key characteristic of a swarm is that the dynamic agents within the swarm interact strongly in order to enhance the performance of the swarm as a whole. One important aspect

of interaction are the observations by the dynamic agents of the states of the particles. This is closely related to a second form of interaction, namely *communication*. Agents may communicate in order to share information about, e.g., the observations and strategies. The moving agents can communicate in various ways. One way is through their particle state, e.g., by making meaningful movements. This is observed in nature with honey bees. Another way is by changing the environment in a meaningful way through its process state. This is called stigmergy and is observed in nature with ants. A third way is by sending and receiving messages through the *communication channel*  $C$ . In the framework, this is incorporated by the message signals  $\mathbf{v}$  and  $\mathbf{w}$ , the message space  $W$ , and the message generating function  $\phi$ , defined in Definition 2.2.

As the moving agents in a swarm are typically low powered, with short-ranged sensors, they are only capable of communicating with other agents within a certain *neighborhood*. The neighborhood of a moving agent is defined as the set of moving agents that the given agent can receive messages from and is a function of the communication properties of the moving agents and their positions.

*Definition 2.6:* The neighborhood of a moving agent  $M_i$  is denoted by the set  $\mathcal{N}_i(\mathbf{x}_s, \boldsymbol{\sigma})$ , with  $\mathbf{x}_s$  the vector stacking the particle states and  $\boldsymbol{\sigma}$  the vector of communication parameters of all the moving agents respectively.

The neighborhoods of the dynamic agents are dynamic, as they are dependent on the particle states  $\mathbf{x}_s$ . The framework allows for any method to model the communication channel. With limited broadcast power, the neighborhood of a moving agent consists only of the other moving agents that are within a certain radius. The communication parameter  $\boldsymbol{\sigma}$  then consists of these radii. The notion of neighborhood is defined in the framework to be part of the environment, as it depends on the physical properties of the moving agents. Also properties of the process, such as bandwidth of the communication channel and the delay and attenuation of the signals traveling through the communication channel are relevant to this definition.

It must be noted that in order to retain flexibility and scalability of the swarm, the dynamic agents are not able to directly address other agents. The agents thus broadcast their messages and the neighborhood defines which agents receive them.

*Example 7:* The robots broadcast the messages that are constructed by their software. Their position, broadcast power, and the properties of the environment determine their neighborhoods, which dictate which of the other robots receive their messages. The messages can be used to improve an internal world model and decision making.  $\square$

### G. Swarms

Finally the notion of a swarm can be formalized.

*Definition 2.7:* A swarm is a subset of the set of moving agents,  $\mathcal{S}$ , with the dynamic agents being cooperative.

In Definition 2.2, the index  $i$  to the spaces and transition functions could have been added to stress that the framework allows for all the moving agents to have different properties. This gives rise to the notions of *homogeneous* and *heterogeneous* swarms.

*Definition 2.8:* If all the moving agents in the swarm have the same state transition functions, the swarm is said to be homogeneous. If at least one of the state transition functions is different, the swarm is said to be heterogeneous.

### III. RELATION TO STATE OF THE ART

This section relates the proposed framework to the two major swarm intelligence methods in optimization and control, namely Particle Swarm Optimization (PSO) and artificial potential fields for swarm aggregation.

#### A. Particle Swarm Optimization

PSO [13] is an optimization heuristic, in which the goal is to find the parameter vector associated with the global optimum in a problem space according to a certain objective, or fitness function. It is different from most other optimization heuristics as it uses a swarm of agents instead of only one agent. In this way, it is closely related to evolutionary computation, which is also population-based. Similarities and differences are discussed in, e.g., [1]. The agents in PSO are called particles and partly conform with the particles from the framework introduced in this paper. A particle  $i$  is defined by a state  $\mathbf{x}_i = [\boldsymbol{\theta}_i^T \ \mathbf{v}_i^T]^T$  denoting its position and velocity in the problem space  $\Theta$ . A fitness function  $F(\boldsymbol{\theta}) : X \rightarrow \mathbb{R}$  maps the parameter space to a fitness landscape, which as a result associates each particle to a fitness value.

In PSO, the particles evolve in discrete time, because of the algorithmic nature of the optimization problem. In the basic setting, at each iteration of the algorithm, the particles update their state with the following rule:

$$\boldsymbol{\theta}_i(k+1) = \boldsymbol{\theta}_i(k) + \mathbf{v}_i(k), \quad (1)$$

$$\begin{aligned} \mathbf{v}_i(k+1) = & w(k)\mathbf{v}_i(k) + c_1 r_1(k)[\boldsymbol{\theta}_{i,\text{pbest}}(k) - \boldsymbol{\theta}_i(k)] \\ & + c_2 r_2(k)[\boldsymbol{\theta}_{i,\text{lbest}}(k) - \boldsymbol{\theta}_i(k)] \end{aligned} \quad (2)$$

where  $k$  is the current time step,  $\boldsymbol{\theta}_{i,\text{pbest}}$  is the *personal best* position,  $\boldsymbol{\theta}_{i,\text{lbest}}$  is the *local best* position,  $w(k)$  is the inertia weight,  $r_{1,2}(k)$  are random variables, and  $c_{1,2}$  are positive acceleration constants. The personal and local best positions are the values of  $\boldsymbol{\theta}_i(k)$  that are associated with the highest fitness value attained since  $k = 0$  for particle  $i$  and any particle in the neighborhood of that particle respectively.<sup>2</sup> Each particle in the swarm is attracted towards its personal best solution and the local best solution. In this way, it learns to find the optimum of the fitness function, not only by its own experience, but from other members of the swarm as well. The values of the inertia weight  $w(k)$  and the range of the random variables  $r_{1,2}(k)$  influence the convergence properties

<sup>2</sup>Sometimes, the neighborhood is considered to cover the complete swarm. In that case, the local best is called the *global best* position of a particle,  $\boldsymbol{\theta}_{i,\text{gbest}}$ .

of the particle swarm. The positive acceleration constants  $c_{1,2}$  trade off exploration and exploitation. More information can be found in [13] and [14].

The equations (1) and (2) can be written in a state-space form that separates the particle from the dynamic agent according to the framework proposed in this paper:

$$\begin{aligned} \mathbf{x}_i(k+1) = & \begin{bmatrix} 1 & 1 \\ 0 & w(k) \end{bmatrix} \mathbf{x}_i(k) \\ & + \begin{bmatrix} 0 & 0 \\ c_1 r_1(k) & c_2 r_2(k) \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_{i,\text{pbest}} - \boldsymbol{\theta}_i \\ \boldsymbol{\theta}_{i,\text{lbest}} - \boldsymbol{\theta}_i \end{bmatrix} (k). \end{aligned}$$

From this form, the particle  $P_i$  can be identified by the relation:

$$\mathbf{x}_i(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & w(k) \end{bmatrix} \mathbf{x}_i(k) + \mathbf{u}_i(k),$$

with

$$\mathbf{u}_i(k) = \begin{bmatrix} 0 & 0 \\ c_1 r_1(k) & c_2 r_2(k) \end{bmatrix} \mathbf{z}_i(k) \quad (3)$$

the input from the dynamic agent  $A_i$ , having an internal state  $\mathbf{z}_i = [(\boldsymbol{\theta}_{i,\text{pbest}} - \boldsymbol{\theta}_i)^T \ (\boldsymbol{\theta}_{i,\text{lbest}} - \boldsymbol{\theta}_i)^T]^T$ .

The process represents the fitness function  $F(\boldsymbol{\theta})$  and the observation signal is defined as  $\mathbf{y}_i = [F(\boldsymbol{\theta}_i) \ \boldsymbol{\theta}_i^T]^T$ . Each dynamic agent determines its personal best position as  $\boldsymbol{\theta}_{i,\text{pbest}}(k) = \arg \max (F(\boldsymbol{\theta}_i(k)), F(\boldsymbol{\theta}_i(k-1)))$  and broadcasts it as the message  $\mathbf{v}_i(k) = [F(\boldsymbol{\theta}_i) \ \boldsymbol{\theta}_{i,\text{pbest}}(k)^T]^T$ . The communication channel determines the neighborhood of each moving agent and produces the message  $\mathbf{w}_i = \{F(\boldsymbol{\theta}_j), \boldsymbol{\theta}_{j,\text{pbest}} \mid M_j \in \mathcal{N}_i(\mathbf{x}_s, \sigma)\}$ , with  $\sigma$  the size of the neighborhood, equal for all the moving agents. Dynamic agent  $A_i$  processes this message by taking the maximum over the fitness values in this set to determine the local best position  $\boldsymbol{\theta}_{i,\text{lbest}}$ . This provides the dynamic agent with enough information to update the internal state  $\mathbf{z}_i$  and produce its input to the environment  $\mathbf{u}_i$  according to (3). Here, the stochastic nature of the decision making is expressed by the random variables  $r_{1,2}(k)$ .

#### B. Controlling a Particle Swarm

One of the main control problems studied in literature is *swarm aggregation*, in which the agents have to aggregate to form a cohesive swarm [15]. For analyzing the swarm behavior, most of the research has focused on a simple model of the particle dynamics and their interaction. The particles are modeled by a kinematic model:

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t), \quad (4)$$

where the position of a particle  $i$  at time  $t$  is denoted by  $\mathbf{x}_i(t)$  and its corresponding input by  $\mathbf{u}_i(t)$ . This model allows proof-of-concept design of swarm systems, where at a later stage (4) can be replaced by a more realistic, more complex model, like a point mass model or full actuator model [15], [16]. The input to the particle dynamics is the local value of an artificial potential field. In the environment, all the objects, such as the particles and obstacles are assigned a *potential*

function that defines a virtual force acting upon a particle at a certain distance. The value of the artificial potential field is the sum of the values of all the potential functions. The general class of attraction/repulsion functions studied in [17] is of the type:

$$g(\mathbf{y}) = -\mathbf{y}[g_a(\|\mathbf{y}\|) - g_r(\|\mathbf{y}\|)], \quad (5)$$

where  $g_a, g_r : \mathbb{R}^+ \rightarrow \mathbb{R}^+$  represent the magnitude of the attraction and repulsion term respectively, the vector  $\mathbf{y} = \mathbf{x}_i - \mathbf{x}_j$  represents the distance between two particles  $i, j \in \{1, \dots, M\}$ , and  $\|\mathbf{y}\| = \sqrt{\mathbf{y}^T \mathbf{y}}$  is the Euclidean norm. The input  $\mathbf{u}_i$  is generated by the dynamic agent  $A_i$  based on measurement of the distance between its own particle and other particles in the environment,  $\mathbf{y}_i = \{\mathbf{x}_i - \mathbf{x}_j \mid M_j \in \mathcal{N}_i(\mathbf{x}_s, \sigma_i)\}$ . The neighborhood is defined by the set  $\mathcal{N}_i(\mathbf{x}_s, \sigma_i) = \{M_j \mid \|\mathbf{x}_i - \mathbf{x}_j\| < \sigma_i\}$ , with  $\sigma_i$  the sensing radius of  $P_i$ . The function  $g$  is typically predefined and identical for all the moving agents. Dynamic agents may also infer  $g$  from observed physical properties of other moving agents. The color of a moving agent, e.g., may be associated with a certain  $g$  according to a prespecified database. Parameters that define  $g$  may also be communicated by the signals  $\mathbf{v}$  and  $\mathbf{w}$ .

#### IV. CONCLUSIONS AND OUTLOOK

This paper has introduced a general framework for swarm systems. It is an essential first step before any proper development and analysis of more intelligent swarms can be done. The proposed framework provides a good basis for future research.

- The framework enables a more structured approach to the development of new applications of swarm intelligence, particularly for distributed sensing and control.
- It provides a better insight in the structure of swarm systems. Theoretical results and results based on simulation can be more directly mapped to the application domain.
- The framework separates the physical parts and behavior of the swarm members from their decision making capabilities. This facilitates the integration of current swarm intelligence research, which focuses mainly on the physical behavior of the swarm members, with research on more sophisticated decision making in dynamic agent systems and AI. The proposed framework aims to integrate both fields to enable the development and analysis of more sophisticated swarm systems.

Two of the most established methods from the swarm community, PSO and artificial potential functions for swarm aggregation have been related to the proposed framework. Although PSO is an optimization method and swarm aggregation is a control problem, it has been demonstrated how these methods can both be decomposed into similar elements. Future research of the authors will focus on the development and analysis of swarms of moving agents for distributed sensing and control, based on the proposed framework.

#### REFERENCES

[1] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001.

[2] A. Okubo, "Dynamical aspects of animal grouping: swarms, schools, flocks, and herds," *Advances in Biophysics*, vol. 22, pp. 1–94, 1986.

[3] D. Grünbaum, "Schooling as a strategy for taxis in a noisy environment," *Evolutionary Ecology*, vol. 12, no. 5, pp. 503–522, July 1998.

[4] X. Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Oakland, CA, USA, August 2001, pp. 625–630.

[5] Y. Wang and Z. Han, "Stability of an automated vehicle platoon," in *Proceedings of the 1998 American Control Conference (ACC 1998)*, vol. 2, Philadelphia, PA, USA, June 1998, pp. 950–954.

[6] J. T. Feddema, C. Lewis, and D. A. Schoenwald, "Decentralized control of cooperative robotic vehicles: theory and application," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 852–864, October 2002.

[7] C. R. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robotics and Autonomous Systems*, vol. 30, no. 1-2, pp. 85–101, January 2000.

[8] C. J. Augeri, K. M. Morris, and B. E. Mullins, "Harvest: a framework and co-simulation environment for analyzing unmanned aerial vehicle swarms," in *Proceedings of the Military Communications Conference (MILCOM 2006)*, Washington, DC, USA, October 2006, pp. 1–7.

[9] R. W. Deming and L. I. Perlovsky, "Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors," *Information Fusion*, vol. 8, no. 3, pp. 316–330, July 2007.

[10] S. Kornienko, O. Kornienko, and P. Levi, "Minimalistic approach towards communication and perception in microbotic swarms," in *Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2005)*, Edmonton, Canada, August 2005, pp. 2228–2234.

[11] M. Dorigo, V. Trianni, E. Sahin, T. H. Labella, R. Gross, G. Baldassarre, S. Nolfi, F. Mondada, J. L. Deneubourg, D. Floreano, and L. M. Gambardella, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots, Special Issue on Swarm Robotics*, vol. 17, no. 2-3, pp. 223–245, 2004.

[12] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ: Prentice Hall, 1994.

[13] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Western Australia, November 1995, pp. 1942–1948.

[14] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, February 2002.

[15] V. Gazi and K. M. Passino, "Stability of a one-dimensional discrete-time asynchronous swarm," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 35, no. 4, pp. 834–841, August 2005.

[16] V. Gazi and B. Fidan, *Coordination and control of multi-agent dynamic systems: Models and approaches*, ser. Lecture Notes in Computer Science. Springer-Verlag, 2007, vol. 4433, ch. Swarm Robotics: SAB 2006, pp. 71–102.

[17] V. Gazi and K. M. Passino, "A class of attractions/ repulsion functions for stable swarm aggregations," *International Journal of Control*, vol. 77, no. 18, pp. 1567–1579, December 2004.