Sensors **2011**, *11*, 11372-11389; doi:10.3390/s111211372

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Improving Social Odometry Robot Networks with Distributed Reputation Systems for Collaborative Purposes

David Fraga [1,⋆], Álvaro Gutiérrez [2], Juan Carlos Vallejo [1], Alexandre Campo [3] and Zorana Bankovic [1]

[1] Departamento de Ingeniería electrónica, ETSI Telecomunicación, Universidad Politécnica de Madrid, Av. Complutense, 30, 28040 Madrid, Spain; E-Mails: jcvallejo@die.upm.es (J.C.V.); zorana@die.upm.es (Z.B.)

[2] Tecnologías Especiales Aplicadas a la Telecomunicación, ETSI Telecomunicación, Universidad Politécnica de Madrid, Av. Complutense, 30, 28040 Madrid, Spain; E-Mail: aguti@etsit.upm.es

[3] IRIDIA, CoDE, Université Libre de Bruxelles, Av. F. Roosevelt, 50, CP 194/6, 1050 Brussels, Belgium; E-Mail: alexandre.campo@ulb.ac.be

⋆ Author to whom correspondence should be addressed; E-Mail: dfraga@die.upm.es; Tel.: +34-915-495-700 ext. 4227; Fax: +34-913-367-323.

Received: 31 October 2011; in revised form: 21 November 2011 / Accepted: 22 November 2011 / Published: 30 November 2011

Abstract: The improvement of odometry systems in collaborative robotics remains an important challenge for several applications. Social odometry is a social technique which confers the robots the possibility to learn from the others. This paper analyzes social odometry and proposes and follows a methodology to improve its behavior based on cooperative reputation systems. We also provide a reference implementation that allows us to compare the performance of the proposed solution in highly dynamic environments with the performance of standard social odometry techniques. Simulation results quantitatively show the benefits of this collaborative approach that allows us to achieve better performances than social odometry.

Keywords: collaborative robots; robot networks; social odometry; collective decision; reputation systems; trust algorithms; unsupervised techniques

---

1. Introduction

This paper proposes a new approach for collaborating purposes in a swarm of robots working together to achieve a goal. Robots are individual sensors highly efficient, equipped with sufficient abilities, that can be exploited jointly. The collaborative swarm is a group of entities that work together to achieve a common objective. They make intelligent decisions to achieve a foraging goal which requires some mechanism of collaboration by means of social odometry. In social odometry, each robot is a sensor for the other robots of the swarm. The importance of social odometry lies on the fact that the swarm (the collectivity) allows the robots to collaborate to achieve a common objective because the individuals are working together.

Many robotics applications require the robots to be localized to achieve different tasks. Different solutions to the localization problem have been implemented. Among these, odometry is probably the most used as it provides easy and cheap real time position information by the integration of incremental motion information over time. Unfortunately, this integration causes an accumulation of errors during the movement of the robot, and this can be a great drawback in some robotic applications, such as foraging, where the robots have to find, select and exploit resources from unknown locations.

Different approaches have been implemented to deal with this complexity; however, those solutions have a number of different limitations: (i) they are power consuming in terms of computation [1,2]; (ii) some robots are not allowed to move or they have its mobility limited [3]; (iii) robots must maintain visual contact at all times with the rest of the group [4]; and (iv) in some cases robots have to communicate with a central device to update or download maps of their environment, synchronize movements, or update positions [5].

Social odometry [6,7] is a novel solution that exploits self-organized cooperation in a group of robots to reduce each individual location error. Each robot location knowledge consists of an estimate of its own location and an associated confidence level that decreases with the distance traveled since the last known location. In order to maximize its confidence about its estimate, each individual tries to update it by using the information available in its neighborhood. Estimated locations, confidence levels and actual locations of the robots co-evolve in parallel in order to guide each robot to the correct objective.

In this paper, we work with a classical swarm foraging scenario: a number of resource items (usually called "prey") are randomly scattered in the arena. In this context, robots search and retrieve those resource-items back to a specific place (usually called "nest"). The performance of the robot network in this kind of foraging systems can be measured as either the resources-items collected by unit of time, or the time robots need to exhaust the resources.

As aforementioned, social odometry uses a simple reputation system based on the distance traveled. However, from the point of view of reputation systems techniques, foraging scenarios have more useful trust information sources that have not been used in previous works [6,7].

In this paper, we state that by defining a complete architecture and following a systematic reputation-system analysis and design processes, it is possible to improve the performance of social odometry. Hence, we propose a complete reputation-system architecture and an analysis and design methodology, and provide a reference implementation that allow us to compare the performance of the proposed approach with the performance of social odometry.

The rest of this paper is organized as follows: Section 2 explains how social odometry works. In Section 3 we provide a brief introduction to the main topics related to reputation systems and provide a reference architecture. Section 4 analyzes in detail how reputation systems can improve social odometry robot networks. In Section 5 we present the experimental results. Finally, in Section 6 we draw some conclusions.

2. Social Odometry

2.1. The Odometry Problem

Odometry is probably the most used localization method. It provides easy and cheap real time position information through the integration of incremental motion information over time without the need for any other device. In all odometry techniques, a travel path is derived from sensors computing the movement of the robot. However, the accuracy of odometry measurements strongly depends on the kinematics of the robot. Unfortunately, because of the integration of the robot's movement, odometry calculation causes an accumulation of errors, where problems such as slippage, misalignment of the wheels or several other inaccuracies must be taken into account [8].

Odometry errors can be classified as either systematic or non-systematic errors [9]. Systematic errors can be modeled and corrected, while the non-systematic ones cannot be corrected and many classical techniques have been implemented to cope with them.

2.2. Learning from Others

Social odometry is a previously defined technique [7,10] which is not based on any map-like algorithm, and despite being inspired by the Kalman Filter [6,11], it does not require any explicit model of the movement errors. On the contrary, a relationship between the distance traveled and a confidence level allows the robots to select the closest resource site on a foraging-like scenario.

The key aspect of social odometry is that robots within the swarm act as virtual landmarks to the others and exchange their knowledge about the position of goal areas. Nonetheless, they have to deal with two main issues: (i) the robots only know estimated locations, not the real locations; and (ii) the more the robots travel the worse those estimates are.

To comply with the aforementioned characteristics, social odometry uses a range and bearing communication sensor [12,13] which provides a local, distributed and situated communication. This sensor allows the robots to obtain the information transmitted by their neighbors, as well as the range and bearing to the emitting source. The communication does not rely on any central unit. Moreover, no synchronization is needed by the robots to exchange their information, removing the need for a common time axis. However, because the robots do not have any inertial system, the sole common coordinate system lies on the range and bearing communication system.

2.3. Social Odometry Equations

In social odometry, we define the state vector of the robot i at time k as:

$$\boldsymbol{x}_k^i = \begin{bmatrix} x_k^i & y_k^i & \theta_k^i \end{bmatrix}^T \tag{1}$$

where x_k^i and y_k^i are the robot's Cartesian coordinates and θ_k^i its orientation.

Moreover, the inverse of the confidence level (p_k^i) is defined as distance travelled by the robot (d_k^i).

Every robot keeps track of its movements and updates its *a priori* estimated location and confidence level about the different goals (*i.e.*, nest and prey) as:

$$\begin{aligned} \widehat{\boldsymbol{x}}_{k|k-1}^{goal,i} &= \widehat{\boldsymbol{x}}_{k-1|k-1}^{goal,i} + \Delta\widehat{\boldsymbol{x}}_k^i \\ p_{k|k-1}^{goal,i} &= p_{k-1|k-1}^{goal,i} + \Delta d_k^i \end{aligned} \tag{2}$$

where $\Delta\widehat{\boldsymbol{x}}_k^i$ is the state vector displacement in the time step duration and Δd_k^i is the distance travelled in the time step duration.

If there is no encounter between the robots, the *a posteriori* values are matched to the *a priori* values ($\widehat{\boldsymbol{x}}_{k|k}^{goal,i} = \widehat{\boldsymbol{x}}_{k|k-1}^{goal,i}$, $p_{k|k}^{goal,i} = p_{k|k-1}^{goal,i}$). Therefore, the confidence level decreases indefinitely. On the other hand, if two robots meet, the robots exchange information about their position and confidence level. In order to produce an *a posteriori* estimated location, each robot takes into account all information available, but weighs its sources in a different way:

$$\widehat{\boldsymbol{x}}_{k|k}^{goal,i} = \left(1 - g_k^{goal,i}\right) \widehat{\boldsymbol{x}}_{k|k-1}^{goal,i} + g_k^{goal,i} \left(\widehat{\boldsymbol{x}}_{k|k-1}^{goal,j} + \boldsymbol{x}_k^{ij}\right) \tag{3}$$

$$p_{k|k}^{goal,i} = \left(1 - g_k^{goal,i}\right) p_{k|k-1}^{goal,i} + g_k^{goal,i} p_{k|k-1}^{goal,j} \tag{4}$$

where $\boldsymbol{x}_k^{ij}$ is the vector from one robot i to robot j and g_k represents the so-called pairwise comparison rule often adopted in evolutionary/social dynamics studies [14], to code the social learning dynamics, which makes use of the Fermi distribution:

$$g_k^{goal,i} = \frac{1}{1 + e^{-\beta\left(\Delta p_{k|k-1}^{goal,ij}\right)}} \tag{5}$$

where $\Delta p_{k|k-1}^{goal,ij} = p_{k|k-1}^{goal,i} - p_{k|k-1}^{goal,j}$ and β measures the importance of the relative confidence levels in the decision making.

Therefore, social odometry fuses the robot estimations based on their confidence levels. An exhaustive revision of the social odometry equations can be found in [11].

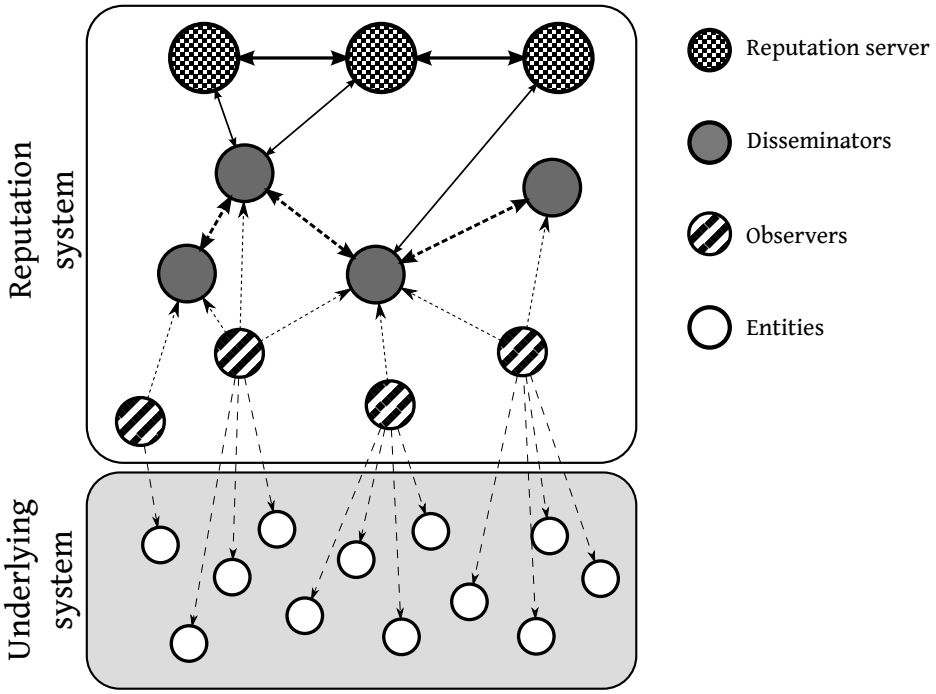
3. Reputation Systems

Trust and reputation have recently been suggested as an effective security mechanism for open and distributed environments (*Ad Hoc* networks, WSNs, P2P networks, *etc.*). Extensive research has been done on modeling and managing trust and reputation. Specifically, it has been demonstrated that rating trust and reputation of individual nodes is an effective approach in distributed environments not only to improve security, but to support decision-making and promote node collaboration.

There are many different definitions of trust and reputation [15]. In essence trust is a belief about future behavior that one participant in the system holds in others and it is based on its own experience, thus its main characteristic is subjectivity. On the other hand, reputation is considered to be the global perception of the behavior of a node based on the trust that others hold in it. Thus, reputation is considered to be objective.

In order to identify the fundamental entities of reputation systems, an architectural model for reputation systems is presented (see Figure 1). Therefore, we can analyze all the entities involved in a trust/reputation dynamic and all the processes needed to effectively take advantage of this kind of systems.

Figure 1. Generic reputation system architecture.



- **Underlying System**. Reputation systems exist to improve the performance of another system in a specific way. This system is called *underlying system* and its basic components are called *entities*.
- **Observers**. They are the basic agents of the reputation system. They create and manage the trust used by the whole system.
- **Trust Information Sources**. In order to create a useful value of trust for entities, observers can use any of these sources: they can obtain information by direct observation of the *real world*; they can use their *memory*, so they are able to evaluate the historical behavior of the entities; they can use information provided by other *observers* (*communication*); they can use *categorization* as trust source information when the group the entities belong to is associated to a specific trust environment (this is very common in social interactions where it is called *prejudices*); and finally they can use the *reputation* value of the entities (this is common in early interactions or when the global perception of an entity is more important than the local perception).
- **Trust Algorithm**. In order to create a useful value of trust, observers process all or some of the aforementioned sources of information with an internal algorithm. This is a key element in the whole reputation system so it has to be chosen very carefully as we will see in the next sections.
- **Disseminators**. Trust information calculated by *observers* can be used by other observers or can be used to calculate reputation values. In order to allow this transmission of information, some agents within the reputation system can have the capacity of relaying trust information messages.

- **Dissemination Protocol**. Transmission of trust and reputation information carried out by the *disseminators* is based on the existence of a specific communication protocol that is commonly called *dissemination protocol*.
- **Reputation Servers**. Some special agents in the reputation system (or even none or all the agents) can use the trust information generated and distributed by the *observers* and *disseminators* to generate values of *reputation* for all the entities.
- **Reputation Algorithm**. In order to create a useful value of reputation, *reputation servers* use an internal algorithm.

Besides of analyzing these architectural elements, we should take into account how the reputation system is conditioned by the underlying system.

- **Topology**. Related to the dissemination protocol we find that the topology of the underlying system is a key factor. We can find as many topologies as in a generic distributed system (e.g., client-server, multi-agent systems, ad-hoc networks).
- **Timing**. Trust information acquisition, calculation or dissemination are vital processes. The moment *when* they happen can modify and determine the features and effectiveness of the reputation system. The three basic timing schemas are: periodic, event oriented and periodic adaptive.
- **Limitations of the Underlying System**. We must take into account all possible limitations that the underlying system can impose. Among others: communication or computational resources, storage capacity, power consumption, *etc*.
- **Requirements and Goals of the Underlying system**. Reputation systems are a way to improve an underlying system performance in a number of specific criteria. So, the most important task we have to carry out is to identify all these requirements and goals.

4. Reputation Systems in a Social Odometry Context

Social odometry exploits self-organized cooperation in a group of robots to reduce each individual location error using a simple and low-resources-consumption model. This allows us to use this localization technique in a wide range of real-life scenarios. If we could minimize this location error without increasing the complexity order of the solution, we would be able to both improve the performance of social-odometry applications and broaden even more the range of the systems where we can apply social odometry techniques.

As we described before, in order to improve the behavior of basic social odometry techniques we only have to analyze them from the reputation system point of view. In this section we propose and follow a methodology to analyze and design our reputation system.

It is based on three main steps: analyze the underlying system, identify the elements that are going to be part of the reputation system architecture and define how trust and reputation processes (algorithms) are going to be carried out by the system.

4.1. Underlying System Analysis

Based on the aforementioned structure we can identify the following topics about the underlying system.

- **Description of the Underlying System**. Based on a classical social odometry swarm behavior [7], we propose a richer and more complex scenario so we can analyze the viability of this solution in real-life environments. The additional features are: (i) **there are different models of robots** and it is well-known that they have different location performances (some models are better than others); (ii) within a specific model, **individual robots have different location performances** (but this specific performance is not known by the other robots).
- **Requirements and Goals**. Robots have to go to the source of resources ("prey") and go back to the "nest" as many times as they can. This is a paradigmatic example of a *Maximization of the System Performance* scenario.
- **Topology**. There are not central services. The robots have full freedom of movements and all P2P communications between them are allowed if they are near enough.
- **Timing**. There is not a global clock to trigger whole-system behaviors. So the system is event oriented.
- **Limitations**. The main limitations are based on the communication, computational and storage resources of the robots. Power consumption might be a limitation too, but we will not take it into account in this paper.

4.2. Reputation System Analysis

If we review the elements and processes of the proposed reputation system architecture, we can identify the following ones:

- **Observers**. Every robot in the underlying system is a sensor in the network, so it can be an observer in the reputation system.
- **Trust Information Sources**. The main disadvantage of the previous social odometry approach is that it misses some of the traditional trust information sources. They use information from the *real world* (obtained by their sensors) and information from other observers in a simple way (in the P2P robot-to-robot *communications*), however they lack for an accurate use of *memory* and *categorization*. On the one hand, *memory* is a key factor in the system. In the basic social odometry scenario robots only *remember* how long they have been walking since they found a known location. However, a model with more historical information could improve the precision of any trust algorithm. We will see how simple concepts like the global performance of the robot (total distance/number of locations found or number of round-trips done) can significantly increase the throughput of the system. On the other hand, the use of *categorization* can help us to improve the behavior of the system in the early stages. Therefore, robots can have a more accurate knowledge of the confidence level of the positions transmitted by other robots. Even when they have not already had a minimum amount of historical information (*memory*).

- **Trust Algorithm**. Because of the special importance of this matter it will be discussed in detail in the next subsection.
- **Disseminators**. Every robot in the underlying system can act as a disseminator in the reputation system. *Communication* is essential in the social odometry and we will take advantage of it.
- **Dissemination Protocol**. All communications in the system are robot-to-robot communications so we do not need a complex protocol. We only have to deal with physical and link layer issues. Network layer features are not needed.
- **Reputation Server**. Because of the topology of the underlying system and its limitations, there are not any global services, so we will not have a reputation server for the whole system. We could evaluate if all robots or some of them could act as reputation servers, however the concept of reputation would not be realistic in the defined scenario because in this kind of swarms there is not any kind of *a priori* individual knowledge. Besides, we do not have an efficient mechanism to propagate information throughout the network. So, we could not disseminate the reputation values. Anyway, future works could deal with this idea of introducing reputation servers within the system and analyzing advantages and drawbacks of this proposal.
- **Reputation Algorithm**. Based on the previous point, a reputation algorithm is not needed in this scenario.

4.3. The Trust Algorithm. Conceptual Approach and Trust Sources

Based on previous works in social odometry and reputation systems, we will try to define the main requirements of our trust algorithm.

In a system composed of entities with different performance levels, the possibility of having an *a priori* knowledge of this performance or a knowledge of the predictable behavior of these entities can help us to improve the global performance. Optimal filters are a classical approach to this topic, but they are computational expensive compared to the resources available in the robots [16]. However, in a reputation system world, this kind of knowledge is often modeled in a more simple way: the concept of *category*.

Moreover, we have identified that the number of round trips divided by the distance traveled can be a good estimator of the individual performance of every robot in the system.

Finally, the information exchange carried out by the social odometry approach has proved to be valid in this kind of environments. However, it is limited to the transmission of *personal* information. As mentioned before, one of the main trust information sources is carried out by the disseminators, so besides transmitting their own location information, they could transmit trust information about previous known robots based on its individual performances. In this way, trust information could be disseminated faster and the whole system performance might be improved as well.

Based on these previous ideas our trust algorithm will be defined as follows:

The inputs for our algorithm will be: (i) the distance traveled since the last known location, so we can keep the advantages of the classical social odometry approach; (ii) the category or type of robots in the system, so we can introduce an *a priori* knowledge but in a simpler way than using other common techniques (such as Kalman filters [17]); and (iii) the ratio distance divided by number of round-trips, so we will have an estimate of the individual performance.

Moreover, we will store these inputs so we can use this historical information. Finally, we will promote the trust dissemination between robots.

4.4. The Trust Algorithm: Algorithm Specification

In order to implement the algorithm we could have used some standard trust algorithm, such as beta algorithm [18], genetic algorithms [19] or self-organized maps [20]. However, in this environment none of them suits our requirements. They are computational expensive, so we decided to adapt the Fermi distribution used in social odometry.

Based on this equation, we are going to introduce the main improvements we commented before. First of all, we will introduce the idea of *category*. In our system there will be three kinds of robots based on the accuracy of their location sensors. Respectively, tolerance will be 2%, 5% and 10%. To introduce this concept in the algorithm we will model this tolerance as maximum errors, so the new "confidence level" will be weighted by this error estimation:

$$E_{category,j} = \begin{cases} 0.02, & \text{if tolerance is } \pm 2\% \\ 0.05, & \text{if tolerance is } \pm 5\% \\ 0.10, & \text{if tolerance is } \pm 10\% \end{cases} \tag{6}$$

$$\varepsilon'_j = \varepsilon_j * E_{category,j} = \frac{1}{d_j\,(loc)}\,(1 - E_{category,j}) \tag{7}$$

The next step is to introduce the idea of *memory* in the form of an estimated error. We will use the aforementioned simple ratio: total distance divided by number of round-trips.

Firstly, we define "estimated distance from nest to prey" for an entity i as follows:

$$D_{NP,i} = \frac{T_{length,i}}{N_{rounds,i}} \tag{8}$$

So, the better the performance the shorter the distance.

Then, we define the estimated error of the entity j (observee) from the point of view of the entity i (observer) as given by the next equation:

$$E_{memory,ji} = \begin{cases} \frac{D_{NP,j} - D_{NP,i}}{D_{NP,i}}, & \text{if } D_{NP,j} - D_{NP,i} > 0 \\ 0, & \text{if } D_{NP,j} - D_{NP,i} \leq 0 \end{cases} \tag{9}$$

There are two important ideas we should clarify. Firstly, we have introduced the idea of subjectivity. We remarked "trust" is a subjective concept but we had not yet used this fact: the "confidence level" now depends on the observer. Secondly, related to the Equation (9), we only define $E_{memory,ji} \neq 0$ when the observer has a better performance than the observee. In this way, robots with worse performance cannot say that robots with better individual performances are wrong.

Finally, we can introduce this memory error ratio in our "confidence level" as follows:

$$\varepsilon''_{j,i} = \varepsilon'_j * E_{memory,ji} = \frac{1}{d_j\,(loc)}\,(1 - E_{category,j})\,(1 - E_{memory,ji}) \tag{10}$$

Finally, the dissemination process does not need to be introduced in the algorithm, but in the exchanged information. If robots exchange their D_{NP} tables and their estimates of different locations,

an entity can use those estimates even when it has not had a previous direct communication with other entities. However, this can introduce a significant overload both in storage and computational resources. We will analyze the effects of the trust dissemination in the next section.

5. Experimental Results

5.1. Simulation Tools

The proposed algorithms have been tested in simulation. We used a simulator of robot networks developed by the IRIDIA research group from Université Libre de Bruxelles. This simulation platform is a fast multi-robot simulator for the e-puck robot [21,22]. It has a custom rigid body physics engine, specialized to simulate only the dynamics in environments containing flat terrain, walls and holes. This restriction allows for certain optimization in the computation of the physics and, thereby, reduces the computational resources needed for running simulations (see [23] for more details). This platform has been combined with a high level abstraction layer based on a reputation-system simulator called TRS-SIM, designed and implemented by the DIE research group from Universidad Politécnica de Madrid. TRS-SIM is now under a final revision previous to its public release. However, it has already been successfully used in several scientific works [24–27] related to trust and reputation systems applied to different disciplines.

The robot network simulator is responsible for kinematic, sensing, decision making and communication tasks while the logic of the reputation system simulator is responsible for the trust generation and management and provides high level information for the decision-making module of the robot network simulator. The combination of these two specific simulators allow us to derive novel results in this area of knowledge.

In our simulations, a robot is modelled as a cylindrical body of 3.5 cm in radius that holds 8 infrared proximity sensors distributed around the body, 3 ground sensors on the lower-front part of the body and a range and bearing communication sensor. IR proximity sensors have a range of 5 cm, while the range and bearing sensor used for the communication has a range of 15 cm. For the three types of sensors, we have sampled real robot measurements and mapped the data into the simulator. Furthermore, we added uniformly distributed noise to the samples in order to simulate effectively the different sensors. Up to $\pm 20\%$ noise is added to the infrared sensors and up to $\pm 30\%$ to the ground sensors. In the range and bearing sensor, noise is added to the range (up to ± 2.5 cm) and bearing (up to $\pm 20^{\circ}$) values. Moreover, each message emitted can be lost with a probability that varies linearly from 1% when the sender-receiver distance is less than 1 cm, to 50% when the two robots are 15 cm from each other. A differential drive system made up of two wheels is fixed to the body of the simulated robot. Errors have also been introduced into the encoder sensors chosen uniformly random in $\pm 20\%$ of the maximum movement at each time step for each wheel.

5.2. Simulation Experiment

In this section, we compare results obtained for different social odometry experiments with the ones obtained for the proposed reputation system scheme based on all the analysis and design decisions followed in the previous sections. Experiments have been tested in a typical foraging scenario. The

selection of this scenario has been made in order to allow for comparison with previous social odometry experiments. However, an extension and generalization of the social odometry algorithms is suggested in Section 6.

It is important to notice that typical social odometry experiments assume that all the robots in the swarm are homogeneous. We have already defined in Section 3 that reputation systems are able to improve the swarm behavior even if the robots are heterogeneous (e.g., differences in the fabrication process). Therefore, all the experiments presented in this section assume the swarm is made up of three categories of robots related to the fabrication process.

Based on the previous assumptions different experiments have been implemented:

- **No Odometry Error**: robots in the swarm do not have odometry error. Therefore, they navigate with a precise knowledge about the goals location ($\widehat{\boldsymbol{x}}_k^{goal,i} = \boldsymbol{x}_k^{goal,i}$, $p_k^i = 0; \forall k, i$)
- **Homogeneous Covariance Knowledge**: robots implement a Kalman Filter to fuse their own information and the one provided by their neighbor. In these experiments, the robots need to calculate the Kalman gain every time step. Because of the comparison with previous works, all the robots assume they have the same noise on both the kinematic and communication for the Kalman Filter equations. Moreover, each robot transmits its estimated location and its own a posteriori covariance matrix when it meets with other neighbors.
- **Social Odometry**: robots communicate using the social odometry filter presented in Section 2.3. In these experiments the robots only transmit their estimated location and confidence level (inverse to the distance traveled).
- **Heterogeneous Covariance Knowledge**: robots uses a Kalman Filter to fuse their own information and the one provided by their neighbor. As in the homogeneous covariance knowledge, the robots need to calculate the Kalman gain every time step. In this experiment the estimated noise is based on the category of the robots involved.
- **Advanced Reputation System—Category**: robots use the proposed reputation system. The trust algorithm only uses the *category* improvement described before (based on the Equation (7)). They must transmit their estimated location, the confidence level and a value based on the quality of their fabrication process.
- **Advanced Reputation System—Memory**: robots use the proposed reputation system. The trust algorithm uses both *categorization* and *memory* as new improvements (based on the Equation (9)). Moreover, they transmit their estimated location, the confidence level, a value based on the quality of their fabrication process and an average value of reliability based on their previous performance.
- **Advanced Reputation System—Dissemination**: robots use the proposed reputation system and disseminate trust information to other robots. So, they transmit their estimated location, the confidence level, a value based on the quality of their fabrication process, an average value of reliability based on their previous performance and a set of average values based on previous communications with other robots.

Finally, the simulations were carried out in a 3×3 m ^{2} and a 5×5 m^2 arenas with two marked areas ("prey" and "nest"), and 30 robots were involved in every experiment. To obtain significant statistical data, the simulations sets were performed one thousand times each.

5.3. Computation and Communication Complexity

5.3.1. Computation Complexity

As aforementioned, covariance knowledge experiments make use of Kalman Filters. The covariance matrix $\boldsymbol{P}_{k|k-1}$ is updated based on the previous *a posteriori* estimated covariance matrix ($\boldsymbol{P}_{k-1|k-1}$) and the noise $\boldsymbol{v}_{k-1}$ through its covariance matrix $\boldsymbol{Q}_{k-1}$:

$$\widehat{\boldsymbol{x}}_{k|k-1} = f\left(\widehat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}, 0\right) \tag{11}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{V}_k \boldsymbol{Q}_{k-1} \boldsymbol{V}_k^T \tag{12}$$

where, $\boldsymbol{A}_k$ and $\boldsymbol{V}_k$ are the Jacobians of $f(\cdot)$ with regard to $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ respectively, and $\boldsymbol{P}_0 = 0$.

On the other hand, in the social odometry, the prediction stage is directly related to the confidence level. Since the spectral norm of the covariance matrix $\boldsymbol{P}$ grows endlessly until a communication is established or the robots arrive at one of the goals, we define the inverse of the *a priori* confidence level ($p^i_{k|k-1}$) of robot i as the distance travelled (d^i_k) since the robot left a specific area. Therefore the prediction stage for the induced covariance matrix is defined as:

$$p^i_{k|k-1} = d^i_k \tag{13}$$

This implementation allows the robot not to calculate the covariance matrix at each time step, and therefore to save computational time.

Moreover, in the covariance knowledge experiments, the correction stage transforms the *a priori* estimated state ($\widehat{\boldsymbol{x}}_{k|k-1}$) into the *a posteriori* estimated state $\widehat{\boldsymbol{x}}_{k|k}$. The *a posteriori* estimated state ($\widehat{\boldsymbol{x}}_{k|k}$) is adjusted in proportion to the Kalman gain ($\boldsymbol{K}_k$), which specifies the degree to which the *a priori* estimation and the measurement $\boldsymbol{z}_k$ are incorporated into the *a posteriori* state. Finally, the *a posteriori* covariance matrix $\boldsymbol{P}_{k|k}$ is also adjusted based on the Kalman gain.

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{W}_k \boldsymbol{R}_k \boldsymbol{W}_k^T\right)^{-1} \tag{14}$$

$$\widehat{\boldsymbol{x}}_{k|k} = \widehat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \left(\boldsymbol{z}_k - h\left(\widehat{\boldsymbol{x}}_{k|k-1}, 0\right)\right) \tag{15}$$

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\right) \boldsymbol{P}_{k|k-1} \tag{16}$$

where, $\boldsymbol{H}_k$ and $\boldsymbol{W}_k$ are the Jacobians of $h(\cdot)$ with regard to $\boldsymbol{x}_k$ and $\boldsymbol{w}_k$ respectively.

Once again, because of the simplification of the covariance knowledge on the social odometry experiments we define g as the scalar value representative to the Kalman gain:

$$g^i_k = \frac{1}{1 + e^{-\beta\left(\Delta p_{k|k-1}\right)}} \tag{17}$$

Hence, we use a weighed average to obtain the new location $\widehat{\boldsymbol{x}}^i_{k|k}$ and the inverse of the confidence level $p^i_{k|k}$ using the Fermi function:

$$\widehat{\boldsymbol{x}}^i_{k|k} = \left(1 - g^i_k\right) \widehat{\boldsymbol{x}}^i_{k|k-1} + g^i_k \left(\widehat{\boldsymbol{x}}^j_{k|k-1} + \boldsymbol{x}^{ij}_k\right) \tag{18}$$

$$p^i_{k|k} = \left(1 - g^i_k\right) p^i_{k|k-1} + g^i_k p^j_{k|k-1} \tag{19}$$

Therefore, it is observed that social odometry implementations are based on scalar values calculations, while covariance knowledge experiments make use of matrices.

5.3.2. Communication Complexity

Because robots in our experiments are used as the measurement $\boldsymbol{z}_k$ to correct the estimates, the estimated state and error needs to be transferred between the robots. In all experiments, robots transmit the *a priori* estimated state ($\widehat{\boldsymbol{x}}_{k|k-1}$), but differences come up with the estimated error communication. In the covariance knowledge experiments robots need to transmit the *a priori* covariance matrix ($\boldsymbol{P}_{k|k-1}$) while in the social odometry robots only transmit scalar values. Table 1 shows a comparison about the information transmitted between the individuals. A maximum of three scalar values is transmitted in all social odometry experiments, with the exception of the dissemination experiment, which depends on the size of the set which must be transmitted. However, as aforementioned, this increase in the communication load is balanced thanks to the reduction on the computation complexity.

Table 1. Information transmitted between the robots when encounter occurs. $\widehat{\boldsymbol{x}}^i_{k|k-1}$ is the *a priori* estimated state, d^i_k is the inverse of the confidence level (distance traveled), q^i_k is the associated quality to the fabrication process, $\bar{r}^i_k$ is the average value of reliability based on their previous performance and r^s_k represents the set of average values based on previous communications with other robots.

Experiment	**Information transmitted**
Covariance knowledge	$\widehat{\boldsymbol{x}}^i_{k\|k-1}$, $\boldsymbol{P}^i_{k\|k-1}$
Social odometry	$\widehat{\boldsymbol{x}}^i_{k\|k-1}$, d^i_k
RS category	$\widehat{\boldsymbol{x}}^i_{k\|k-1}$, d^i_k, q^i_k
RS memory	$\widehat{\boldsymbol{x}}^i_{k\|k-1}$, d^i_k, q^i_k, $\bar{r}^i_k$
RS dissemination	$\widehat{\boldsymbol{x}}^i_{k\|k-1}$, d^i_k, q^i_k, r^s_k

5.4. Results and Discussion

As mentioned before, we carried out two sets of simulations based on the size of the arena (3×3 m^2 and 5×5 m^2). We have implemented the same metric used previously in social odometry experiments, time to elapse the prey, in order to allow comparison with previous works. Results are compiled in Figures 2 and 3.

In the vertical axis we can see a value of performance, meaning by performance the time robots need to exhaust the resources in the "prey". In order to visualize this ratio, we show it in percentage terms compared with the time robots, having no odometry errors, need to exhaust the "prey".

On the other hand, in the horizontal axis, we will display a boxplot for each of the studied odometry techniques (no odometry errors, homogeneous covariance knowledge, basic social odometry, heterogeneous covariance knowledge, improved reputation model based on categorization, improved reputation model based on categorization and memory, and the complete proposed reputation model).

Figure 2. Simulation results for 3 × 3 m^2 arena.

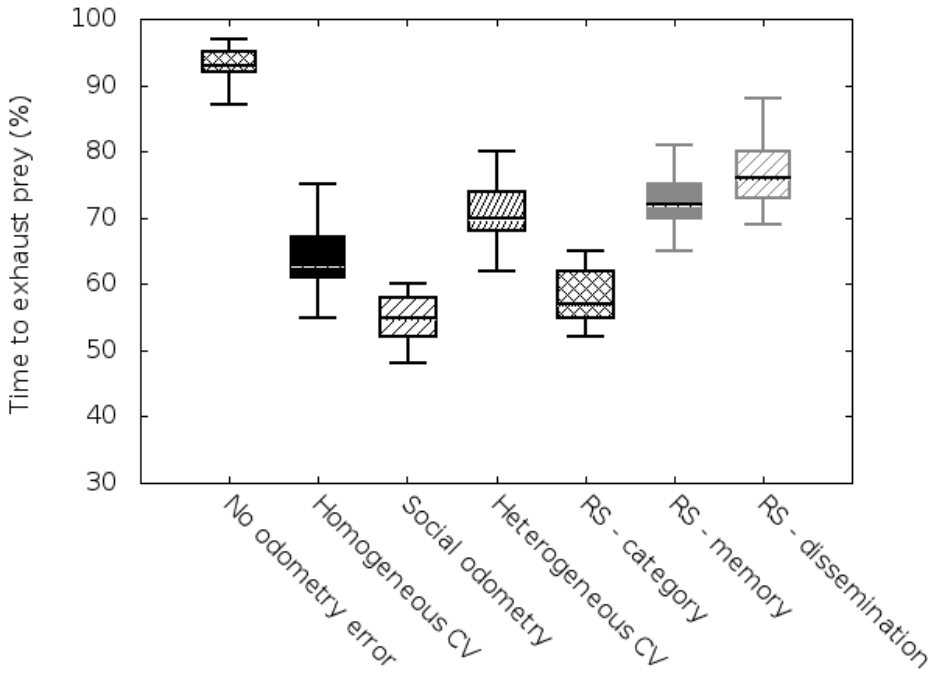
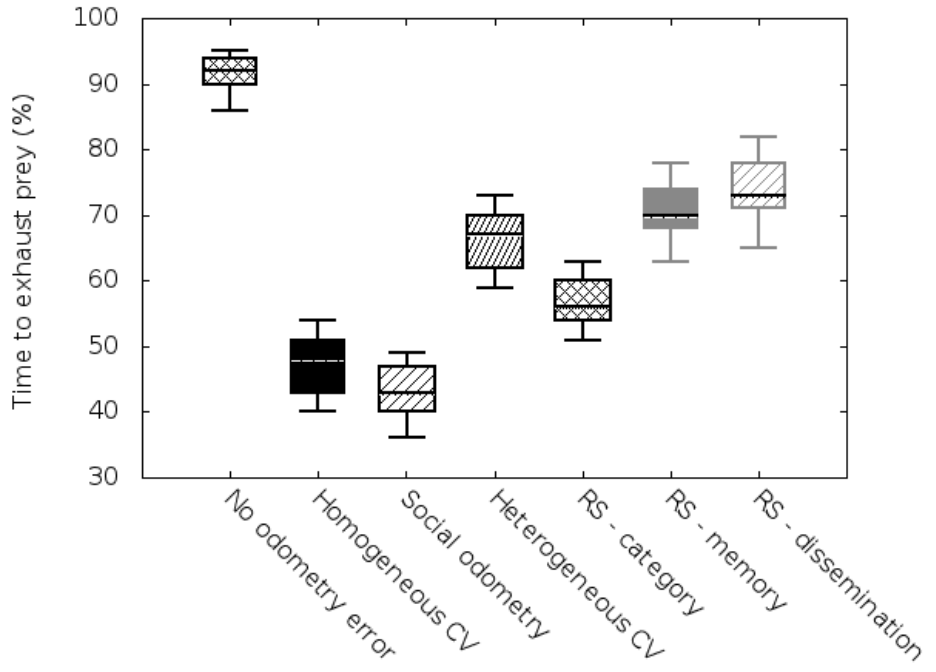


Figure 3. Simulation results for 5 × 5 m^2 arena.



Results of the 3 × 3 m^2 arena are shown in Figure 2. In this case, we can see the results obtained for the basic odometry scenario (no odometry errors, homogeneous covariance knowledge and social odometry) are similar to the results previously obtained in related works [7]. If we analyze the results with *category*-based-reputation system scenario (algorithm based in the Equation (7)), we can observe that the performance obtained in the basic social odometry experiment has been overcome. This difference is because category information helps robots to improve its coordination capabilities in the early stages of the simulation when the swarm is heterogeneous. However, we can see that the heterogeneous covariance

knowledge performance has not been overcome by the *category*-based-reputation experiment. We should not forget that the social odometry approach is a simplification of the covariance knowledge methods.

Anyway, we can find the most important improvement when *memory* is considered and utilized as a trust information source (algorithm based in the Equation (9)). The main difference is because individual performance prevails over local situations (distance traveled since the last know location) and over general statements (categorization). This allow robots to trust more capable entities in the system and follow them as if they were "leaders". In this case, the RS memory experiment shows a similar performance to the heterogeneous covariance knowledge (Wilcoxon test outputs $p \approx 0.5$). It is important to say that this is because robots use more information than in the covariance approach but the improvement is compensated with the model simplification.

Finally, if we take advantage of the trust *dissemination* feature we notice that the results are better than in the heterogeneous covariance knowledge ($p < 0.001$ in the Wilcoxon test). This is because trust information is spread faster and the effect is similar to the use of categorization but with individual information: robots obtain an *a priori* information about the expected individual performance of other robots. Therefore, they can easily trust in the more capable individuals even without previous interactions. However, we have to remember that *dissemination* introduces a significant storage and computational resources overload. So we should evaluate robot's resources in order to know if we can incorporate this technique to our robots.

If we compare these results with the results of the 5×5 m^2 arena scenario (Figure 3), we can see that the reputation system approach offers even better performances. This is because the *a priori* knowledge (categorization) that the robots have helps them to improve their behavior in early stages and this effect is more important in wider scenarios. Without this *a priori* knowledge robots tend to randomly walk around longer throughout the arena and the global performance gets reduced.

Notice that all the experiments, making use of the reputation system, improve previous experiments done with social odometry. The main factor for this improvement is that the robots in the swarm have at hand more information than in standard social odometry algorithms. Therefore, the robots are able to generate a confidence level based not only on their own movement as in standard social odometry but also on the information provided by the other robots in the swarm integrated in time.

6. Conclusions

In this paper we have described how a reputation system can improve the performance of a complex and unsupervised scenario. In order to show it, we reviewed a novel odometry technique, social odometry, and we improved the coordination capabilities of this kind of robot networks designing a reputation system that takes advantage of all the significant information sources we can find in the system. We selected the most suitable trust algorithm and dissemination policies in order to minimize the throughput degradation that less capable robots can induce in the global behavior of the system.

To take advantage of reputation system features we showed the main ideas of a reputation system analysis and design methodology. This methodology is based on the identification of architectural entities, trust and reputation information sources, dissemination algorithms, functional and non functional requirements.

This analysis allowed us to choose the constitutive elements and the more suitable trust algorithms in order to improve the global behavior of a social odometry scenario. Simulation results quantitatively showed that the benefits of this approach were based on the use of *categorization*, *dissemination* and especially *memory*. Since, all of them allowed us to achieve better performances than classical odometry approaches. However, an important drawback could appear with the use of *dissemination*. It requires a significant computational and storage overload in the robots, and this fact can limit its utilization in some real-life scenarios where robots have very few resources. Nonetheless, the resources required during simulation are computationally comparable to the one of the heterogeneous covariance knowledge.

As future work we propose to analyze the viability of introducing a reputation server and a reputation dissemination mechanism within this kind of swarm scenarios. Moreover, a future extension of social odometry should lie on the implementation of general metrics which allow for comparison with other mathematically grounded methods in mobile robotics (e.g., absolute mean error). Besides, the foraging scenario should be generalized and metrics based on the movement error should be extracted. For its implementation an abstract model of the robot and a well-defined random walk algorithm should be extracted in order to allow a concrete comparison between these algorithms.

Acknowledgments

This work was partially funded by the Spanish Ministry of Science and Innovation, under Research Grant AMILCAR TEC2009-14595-C02-01, and the P8/08 within the National Plan for Scientific Research, Development and Technological Innovation 2008–2011. This work was partially supported by the N4C—Networking for Challenged Communications Citizens: Innovative Alliances and Test beds project, funded by the Seventh Framework Program (FP7-ICT-223994-N4C) of the European Commission. Alexandre Campo acknowledges support from the FRS-FNRS and the Jules Reyers Fund of Belgium. The information provided is the sole responsibility of the authors and does not reflect the European Commission's opinion. The European Commission is not responsible for any use that might be made of data appearing in this publication.

References

1. Larsen, T.; Bak, M.; Andersen, N.; Ravn, O. Location Estimation for Autonomously Guided Vehicle Using an Augmented Kalman Filter to Autocalibrate the Odometry. In *Proceedings of the FUSION 98 SPIE Conference*, Las Vegas, NV, USA, 6–9 July 1998; pp. 33–39.
2. Thrun, S.; Burgard, W.; Fox, D. A Real-Time Algorithm for Mobile Robot Mapping With Applications to Multi-Robot and 3D Mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, San Francisco, CA , USA, 24–28 April 2000; pp. 321–328.
3. Grabowski, R.; Navarro-Serment, L.; Paredis, C.; Khosla, P. Heterogeneous teams of modular robots for mapping and exploration. *Auton. Robot.* **2000**, *8*, 293–308.
4. Nouyan, S.; Campo, A.; Dorigo, M. Path formation in a robot swarm: Self-organized strategies to find your way home. *Swarm Intell.* **2008**, *2*, 1–23.

5. Vaughan, R.; Stoy, K.; Sukhatme, G.; Matarić, M. LOST: Localization-space trails for robot teams. *IEEE Trans. Robot. Autom.* **2002**, *18*, 796–812.
6. Gutiérrez, A.; Campo, A.; Monasterio-Huelin, F.; Magdalena, L.; Dorigo, M. Collective decision-making based on social odometry. *Neural Comput. Appl.* **2010**, *19*, 807–823.
7. Gutiérrez, A.; Campo, A.; Santos, F.C.; Pinciroli, C.; Dorigo, M. Social Odometry in Populations of Autonomous Robots. In *Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence, ANTS'08*, Brussels, Belgium, 22–24 September 2008; pp. 371–378.
8. Klarer, P. *Simple 2-D Navigation for Wheeled Vehicles*; Technical report; Sandia Report SAND88-0540; Sandia National Laboratories: Livermore, CA, USA, 1988.
9. Feng, L.; Borenstein, J.; Everett, H. *Where am I? Sensors and Methods for Autonomous Mobile Robot Positioning*; University of Michigan Press: Ann Arbor, MI, USA, 1994.
10. Gutiérrez, A.; Campo, A.; Santos, F.C.; Monasterio-Huelin, F.; Dorigo, M. Social odometry: Imitation based odometry in collective robotics. *Int. J. Adv. Robot. Syst.* **2009**, *6*, 129–136.
11. Gutiérrez, A.; Campo, A.; Monasterio-Huelin, F.; Magdalena, L. Self-Organized Distributed Localization Based on Social Odometry. In *Introduction to Modern Robotics I*; Chugo, D., Yokota, S., Eds.; iConcept Press: Annerley, Australia, 2011; Chapter 1, pp. 1–24.
12. Gutiérrez, A.; Campo, A.; Dorigo, M.; Amor, D.; Magdalena, L.; Monasterio-Huelin, F. An open localization and local communication embodied sensor. *Sensors* **2008**, *8*, 7545–7563.
13. Gutiérrez, A.; Campo, A.; Dorigo, M.; Donate, J.; Monasterio-Huelin, F.; Magdalena, L. Open E-Puck Range & Bearing Miniaturized Board for Local Communication in Swarm Robotics. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, Kobe, Japan, 12–17 May 2009; pp. 3111–3116.
14. Santos, F.C.; Pacheco, J.M.; Lenaerts, T. Cooperation prevails when individuals adjust their social ties. *PLoS Comput. Biol.* **2006**, *2*, 1284–1291.
15. Boukerch, A.; Xu, L.; EL-Khatib, K. Trust-based security for wireless ad hoc and sensor networks. *Comput. Commun.* **2007**, *30*, 2413–2427.
16. Bongard, J.C. Robabilistic Robotics. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005, MIT Press: 647 pages. *Artif. Life* **2008**, *14*, 227–229.
17. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; Technical Report; University of North Carolina at Chapel Hill: Chapel Hill, NC, USA, 1995.
18. Jøsang, A.; Ismail, R. The Beta Reputation System. In *Proceedings of the 15th Bled Electronic Commerce Conference (Bled EC)*, Slovenia, 17–19 June 2002; pp. 41:1–41:14.
19. Banković, Z.; Bojanić, S.; Nieto, O.; Badii, A. Unsupervised Genetic Algorithm Deployed for Intrusion Detection. In *Hybrid Artificial Intelligence Systems*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 132–139.
20. Muñoz, A.; Muruzábal, J. Self-organizing maps for outlier detection. *Neurocomputing* **1998**, *18*, 33–60.
21. École Polytechnique Fédérale de Lausanne. e-puck Website. Available online: http://www.e-puck.org/ (accessed on 24 November 2011).

22. Mondada, F.; Bonani, M.; Raemy, X.; Pugh, J.; Cianci, C.; Klaptocz, A.; Magnenat, S.; christophe Zufferey, J.; Floreano, D.; Martinoli, A. The E-Puck, a Robot Designed for Education in Engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, Castelo Branco, Portugal, 7 May 2009; pp. 59–65.
23. Christensen, A.L. Efficient Neuro-Evolution of Hole-Avoidance and Phototaxis for a Swarm-Bot. DEA thesis TR/IRIDIA/2005-14, Université Libre de Bruxelles, Bruxelles, Belgium, 2005.
24. Banković, Z.; Moya, J.M.; Araujo, A.; Fraga, D.; Vallejo, J.C.; de Goyeneche, J.M. Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps. *Integr. Comput.-Aided Eng.* **2010**, *17*, 87–102.
25. Bankovic, Z.; Fraga, D.; Moya, J.M.; Vallejo, J.C.; Malagón, P.; Araujo, Á.; de Goyeneche, J.M.; Romero, E.; Blesa, J.; Villanueva, D.; Nieto-Taladriz, O. Improving security in WMNs with reputation systems and self-organizing maps. *J. Netw. Comput. Appl.* **2010**, *34*, 455–463.
26. Banković, Z.; Fraga, D.; Moya, J.M.; Vallejo, J.C.; Araujo, Á.; Malagón, P.; de Goyeneche, J.M.; Villanueva, D.; Romero, E.; Blesa, J. Detecting and Confining Sybil Attack in Wireless Sensor Networks Based on Reputation Systems Coupled With Self-Organizing Maps. In *Proceedings of the 6th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI 2010)*, Larnaca, Cyprus, 6–7 October 2010.
27. Bankovic, Z.; Vallejo, J.C.; Malagón, P.; Araujo, Á.; Moya, J.M. Eliminating Routing Protocol Anomalies in Wireless Sensor Networks Using AI Techniques. In *Proceedings of the 3rd ACM Workshop on Artificial Intelligence and Security, (AISec)*, Chicago, IL, USA, 4–8 October 2010; pp. 8–13.