A Bacterial Colony Growth Algorithm for Mobile Robot Localisation

Andrea Gasparri[*], Mattia Prosperi

Dipartimento di Informatica e Automazione,
via della Vasca Navale, 79 - 00146 Roma - Italia.
{gasparri, prosperi}@dia.uniroma3.it
[*] *Corresponding Author*

Abstract

Achieving robot autonomy is a fundamental objective in Mobile Robotics. However in order to realise this goal, a robot must be aware of its location within an environment. Therefore, the localisation problem (i.e.,the problem of determining robot pose relative to a map of its environment) must be addressed. This paper proposes a new biology-inspired approach to this problem. It takes advantage of models of species reproduction to provide a suitable framework for maintaining the multi-hypothesis. In addition, various strategies to track robot pose are proposed and investigated through statistical comparisons.

The *Bacterial Colony Growth Algorithm* (BCGA) provides two different levels of modelling: a *background level* that carries on the multi-hypothesis and a *foreground* level that identifies the best hypotheses according to an exchangeable strategy. Experiments, carried out on the robot ATRV-Jr manufactured by iRobot, show the effectiveness of the proposed BCGA.

1 Introduction

A mobile robot must be able to safely interact with its environment in order to accomplish any task. Therefore, when considering a robotics control architecture, a localisation module should always be included. The aim of this module is to provide reliable pose information for the robot even in the presence of noisy data and unpredictable environmental interactions. Due to the difficulty of obtaining reliable pose information, the localisation problem has been a highly active field of research over the last two decades.

Localisation can be broken down into three different research problems: position tracking, global localisation and kidnap. However, due to the specific nature of these problems, it has been proven difficult to find a general solution for all three.

A widely applied framework describes the localisation problem as a stochastic estimation problem. In fact, by describing the robot pose as a probability distribution (*belief*), the localisation problem can be viewed as an estimation of the belief over the state space of poses, conditioned on the data coming from sensors.

The most famous approach based on this idea is probably the Kalman Filter [11]. It has been applied extensively to solve the *position tracking problem*, i.e., an instance of the localisation problem in which prior knowledge about the initial pose of the robot is available. The Kalman Filter exploits a Gaussian distribution to describe the robot pose, with the advantage of requiring only two parameters: the mode (describing the expected robot pose), and the variance (representing the accuracy of the estimation). However, this simplicity limits the algorithm when the ability to represent the multi-hypothesis is required, as in the *global localisation problem*, i.e., an instance of the localisation problem in which no prior knowledge is provided about the initial robot pose. This problem can be overcome by parallelising methodologies previously introduced to solve the tracking problem. In [2], for instance, several Gaussian distributions have been exploited to represent the probability distribution of the robot in the environment; while in [10] a hybrid localisation method, using multiple Kalman Filters for hypothesis tracking and probability theory for evidence fusion has been proposed.

A different approach, relying on a grid based discretisation of the state space, was introduced in [4]. Here, the key idea is to build a certainty grid map, as explained in [15], then to accumulate the posterior probability in each cell according to the real robot pose and data coming from the sensors. The resulting algorithm provides reliable maps, but suffers from excessive computational overhead [3].

Another approach, based on Monte Carlo Integrations Methods [8], relies on a weighted mass-point discretisation of the probability distribution. Monte Carlo methods can be used to approximate a large number of probability distributions, providing several interesting advantages such as flexibility and parallelisation in execution. However, these methods suffer from the *degeneracy problem*, i.e., the problem of having a majority of samples with negligible weight after few iterations [1]. This phenomenon turns out to be very tricky when facing the *kidnap problem*, i.e., an instance of the localisation problem in which a well-localised robot is suddenly carried to an arbitrary location during its operation [7]. A solution to this problem is to increase the number of samples, but this method suffers from significant computational overhead. A better way to approach the degeneracy phenomenon is to introduce a resampling step in order to augment the diversity among particles [13]. Alternatively, a suitable candidate for the importance function can be devised which minimises the variance of the importance weight conditioned on data [6].

An alternative approach that does not rely on the Bayesian framework has been proposed in [9]. In this paper, a spatially structured genetic algorithm has been conceived. It uses the complex network theory for the population deployment and exploits the properties of several network models, e.g., *small-world* or *scale-free*, to provide an effective exploration of the environment along with good tracking capabilities.

In this paper, we introduce a new biology-inspired approach. The framework, called the *Bacterial Colony Growth Algorithm*, is composed of two different levels of execution: a *background level* and a *foreground level.* The first takes advantage of models of species reproduction to carry on the multi-hypothesis, while the second selects the best hypotheses according to various specialised strategies which are usually problem dependent. Indeed, this modular structure makes the algorithm very adaptive while considering both different scenarios and objectives.

Thus far, an algorithm for solving multimodal optimisation problems and for tracking multiple optima in a dynamic environment using the notion of species has been proposed in [16]. While in this work the authors simply propose a modification of a Particle Swarm algorithm [12], the algorithm proposed here uses models of species evolution as the underlying paradigm for building a new framework able to carry on the multi-hypothesis.

The paper is structured as follows: a brief overview of the required theoretical background is provided in Section 2; the proposed Bacterial Colony Growth Algorithm is described in 3; the problem settings, simulations and experimental results are given in 4 and finally, the conclusion and future perspective are discussed in 5.

2 Theoretical Background

2.1 Robot, Sensors and Environment Modelling

The robot pose can be uniquely determined in an environment by means of the robot position (x, y) and orientation (ϕ). In this paper the unicycle model has been adopted as the kinematic model for the robot. Such a model is described as follows:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-1}, n_{k-1}) \\ &= x_{k-1} + \begin{bmatrix} \cos\tilde{\phi}_{k-1} & 0 \\ \sin\tilde{\phi}_{k-1} & 0 \\ 0 & 1 \end{bmatrix} u_{k-1} + n_{k-1} \end{aligned} \tag{1}$$

where, $x_k = [r_x, r_y, r_\phi]$ is the robot pose at time-step k (*state*), u_{k-1} is the input at time $k-1$ and n_{k-1} is a white zero mean noise at the same time-step. In particular, the system input is $u_k = (\delta s_k, \delta\theta_k)$, where δs_k is the vehicle displacement and $\delta\theta_k$ is the rotation during the sample time interval δt_k, both measured by proprioceptive sensors.

The robot has been equipped with a set of laser rangefinders arranged in a 360°pattern. The related observation model, taking into account the fact that the environment has been described through a set $\mathcal{M}$ of segments, is:

$$\begin{aligned} z_{j,k} &= h(x_k, \mathcal{M}) \\ &= \frac{|a_r l_j^x + b_r l_j^y + c_r|}{|a_r \cos\theta_j + b_r \sin\theta_j|} \end{aligned} \tag{2}$$

where (a_r, b_r, c_r) are the coefficients of the r-th segment and (l_j^x, l_j^y, θ_j) is the configuration of the laser beam detecting the segment in question.

2.2 Models of Species Evolution

The evolution of species has been modelled mathematically with different approaches. Historically, systems of deterministic differential equations suitable for large population dynamics were the first proposed [14] [18] [19]. More recently, in order to model in-vivo reactions such as metabolic or gene regulations, stochastic differential equations [5] and lattice gas automata using Monte Carlo Algorithms [17] have been introduced.

2.2.1 Competitions and Cooperation among Cohabitant Species

The basic model (apart from the Malthusian one [14]) for describing species evolution is the *logistic* model, introduced by Verhulst [18]. In this model, the Malthusian natality factor r is a linear function of the population numerousness $N(t)$

$$f(N(t)) = r - r'N(t) \tag{3}$$

where r and r' are opportune positive constants typical of the population. From the above equation, it follows that the population grows when the natality rate is positive and the numerousness is not too large (below r/r'). Eq. (3) describes an *auto-regulation* linked to the numerousness in the natality process. Introducing eq. (3) in the Malthusian equation $\frac{dN}{dt} = RN(t)$, where $R = r - m$ is the growth rate, m being the mortality factor, the logistic equation is defined as:

$$\frac{dN(t)}{dt} = (r - r'N(t) - m)N(t) = (R - r'N(t))N(t) \tag{4}$$

The ratio $K = \frac{R}{r'}$ is the *carrying capacity*, and the solution is:

$$N(t) = \frac{K}{1 + (\frac{K}{N(0)} - 1)e^{-Rt}} \tag{5}$$

In an evolutionary framework, different species compete for the same resources in order to survive. In other words, the growth of different species is limited by a common factor. Supremacy (survival) of one species over the others is determined by natural selection. The logistic equation previously introduced for a single species can be properly modified to model such competition. For two species, assuming the overall numerousness $((N_1(t)+N_2(t)))$ as the common factor, the following equations can be derived:

$$\frac{dN_1(t)}{dt} = \left(1 - \frac{(N_1(t) + N_2(t))}{K_1}\right) R_1 N_1(t) \tag{6}$$

$$\frac{dN_2(t)}{dt} = \left(1 - \frac{(N_1(t) + N_2(t))}{K_2}\right) R_2 N_2(t) \tag{7}$$

Subsequently, the *predator-prey* was introduced by Volterra and Lotka [19]. Here the authors consider an environment composed by two populations in which predators eat prey.

$$\frac{dH(t)}{dt} = (a - bP(t))H(t) \tag{8}$$

$$\frac{dP(t)}{dt} = (kH(t) - c)P(t) \tag{9}$$

Depending on the constant values, the populations can present different behaviours, including periodic ones.

Competition and *cooperation* can be modelled in a more general framework, where different species are living in the same environment. Consider again a biological system comprised of two populations P_1, P_2 and a limited resource that both populations need. To use the resource, P_1 and P_2 begin to compete. Let's assume that if one population extinguishes, the other one grows according to logistic law $\frac{dP_i(t)}{dt} = a_iP_i(t) - b_iP_i^2(t)$. Moreover, in the cohabitation, there is an encounter term $cP_1(t)P_2(t)$ that has a control effect. The evolution is then described by:

$$\frac{dP_1(t)}{dt} = (a_1 - b_1P_1(t) - c_1P_2(t))P_1(t) \tag{10}$$

$$\frac{dP_2(t)}{dt} = (a_2 - b_2P_2(t) - c_2P_1(t))P_2(t) \tag{11}$$

where a_is are the growth rates, b_is are the *intra*-specific competition coefficients, c_is are *inter*-specific competition coefficients.

In a similar way, a model for the cooperation can be designed in which a population will extinguish if the other one is lacking. In this model, the genetic evolution of species (either by sexual reproduction, i.e. genetic mixtion, or mutation) is not explicitly defined. In order to handle this, several solutions can be presented. One is to allow new speciation and consider different evolved genetic strains of the same species as different competitive-cooperative populations. Another is to introduce correction terms in the reproduction rates as a result of the overall evolution of a species (interpreted as modified replication capacity).

3 The Bacterial Colony Growth Algorithm

A major issue of the global localisation problem is maintaining a set of hypotheses about the robot pose until a reasonable confidence level of estimation is reached. The *Bacterial Colony Growth Algorithm* takes this issue into account. As a result, it provides two levels of modelling:

- The *Background Level* that provides a suitable framework for modelling and carrying on the multi-hypothesis.
- The *Foreground Level* exploits several exchangeable strategies to track the robot pose.

3.1 Background Level: Multi-Hypothesis Modelling

The models of species reproduction introduced above can be effective in describing and maintaining the multi-hypothesis. In this context, a population of hypothetical robots is considered. Each robot is seen as a *bacterium* in a biological environment, say *Escherichia Coli*, which reproduces asexually. One interesting phenomenon observed in the unicellular organisms is the *chemotaxis response*, in which the cellular movement is oriented towards or away from a chemical compound. Mobile bacteria as E. Coli swim towards areas with a higher concentration of nutrient compounds like sugars (attractors) or amino acids, and away from higher concentrations of noxious compounds (repulsors), so that its motivation is similar to that of a particle in a vector field based on a gradient method. Clearly, the environment is composed of different areas characterised by compounds and concentrations which vary over time. Another interesting characteristic of protozoa is that they can form colonies and aggregate in specific regions. Conversely, when no favourable conditions are present, they wait latently for better times to reproduce (*bet-hedging*).

In the mobile robot localisation context, the nutrient areas represent regions where the measurements $\overrightarrow{m}_{r_t}$, provided by the real robot, match with some of the population estimated measurements $\overrightarrow{m}_{p_t}$s (and bad matches define noxious areas). Moreover, the kinematic model allows bacterial movement. At the same time, attractive and repulsive areas change dynamically according to the real robot movements. In the nutrient areas, the bacteria (robots) can reproduce and form colonies (clusters of robot hypotheses), whose growth is limited by the total resources of the environment and by the colony size. Thus, a natural way of maintaining the multi-hypothesis is achieved. In addition, the growth limitation curbs the unbounded growth of the best hypotheses as well as the extinction of other small-medium size colonies.

Specifically, when a bacterium is in a nutrient area, its chances of reproducing and forming a colony are higher while its replication chances are lowered by overpopulation. Moreover, if the nutrient area is shifting somewhere else, the colony first tries to expand slightly (dispersion), then starts to disintegrate if nutriment is no longer available (the environment becomes noxious), as can be clearly observed when a kidnap occurs. Finally, when attractive areas are unavailable or unreachable, the bacteria become latent and stop reproducing, wandering until suitable conditions are found.

The bacterial colony growth algorithm (BCGA) is shown in detail in Algorithm 1. The reproduction policy for each bacterium-robot is driven by both the match with the real robot measurements and the colony density in the local area. In detail, the nutrient or noxious environment condition is described by the formula:

$$f_1(\overrightarrow{m}_{p_{j,t}}, \overrightarrow{m}_{r_t}) = \frac{1}{M}\sum_{i=1}^{M} e^{-\frac{\left(m_{i,p_{j,t}} - m_{i,r_t}\right)^2}{2\sigma^2}} \tag{12}$$

where σ is tuned coherently with the robot measure confidence intervals. The colony density is defined as:

$$f_2(p_j, P) = min\{1, \frac{1}{\nu N}\sum_{i=1}^{N}\left(e^{-\frac{\|p_j - p_i\|^2}{2\sigma^2}}\right)\} \tag{13}$$

where $\|\cdot\|$ is the Euclidean distance between two points, with $\nu \in [0,1]$ and σ controlling the maximum colony size and the spatial radius respectively.

If a bacterium in a determined spatial radius is considered as an individual in a species S_i (colony), the corresponding deterministic differential equation which holds for large populations is:

$$\frac{dS_i}{dt} = f_1(S_i)\left(1 - \left(f_2(S_i, N) + \sum_{k \neq i} f_2(S_k, N)\right)\right) S_i \tag{14}$$

Note that if $f_2(S_i)$ is approximated with $\frac{S_i}{N}$, the logistic law is obtained while the growth is limited by the density and the size of the other colonies, with

Algorithm 1: Bacterial Colony Growth Algorithm

```
Data: $P_t = \{p_{1,t} \dots p_{N,t}\}$
Result: $P_{t+1} = \{p_{1,t+1} \dots p_{N,t+1}\}$
$i = 1$;
while $(i \leq N)$ do
    latency flag $l = TRUE$;
    $j = 1$;
    while $(j \leq N \wedge i \leq N)$ do
        generate $r \in \mathcal{U}[0, 1]$;
        calculate $f_1(\overrightarrow{m}_{p_{j,t}}, \overrightarrow{m}_{robot_t}) = n \in [0, 1]$;
        calculate $f_2(p_{j,t}, P_{t+1}) = d \in [0, 1]$;
        if $(r < n \cdot (1 - d))$ then
            $p_{i,t+1} = reproduction(p_{j,t}, n)$;
            $i = i + 1$;
            $l = FALSE$;
        end
        $j = j + 1$;
    end
    if $(l = TRUE)$ then
        $p_{i,t+1} = betHedging(P_t)$;
        $i = i + 1$;
    end
end
```

$\sum_k S_k \leq N$ as a boundary condition. Here it is assumed that a colony is determined by a small radius in which nutrient conditions and density can be considered constant. If a bacterium reproduces out of this radius, then it is considered either migrating to another colony or forming a new one. The spatial reproduction of a bacterium p depends on the environmental condition: if favourable, the bacterium reproduces in a small neighbourhood; otherwise it migrates according to a normal distribution, whose variance is inversely proportional to the nutriment conditions.

$$reproduction(p) = \begin{cases} x_p = \mathcal{N}(x_p, \frac{\sigma_1}{f_1(p)}) \\ y_p = \mathcal{N}(y_p, \frac{\sigma_1}{f_1(p)}) \\ \vartheta_p = \mathcal{N}(\vartheta_p, \frac{\sigma_2}{f_1(p)}) \end{cases} \tag{15}$$

Note that as $f_1(p)$ approaches zero, the normal distribution tends to the uniform one: the bacterium is randomly dragged, wandering for attractive areas and the bet-hedging strategy is achieved.

This behaviour turns out to be very effective, in particular when a robot is already roughly localised and a kidnap event occurs. In this case, as soon as the hypothesis measurements no longer match the real one, the reproduction rules at the base of the BCGA will provide an immediate response to the kidnap. That is, colonies will start to expand in a Gaussian way with a standard deviation proportional to the matching criteria. Moreover, since a kidnap is a “drastic”event when compared to the most common sensor problems (such as the inability of a laser range-finder to deal with glass walls), the standard deviation will approach zero more quickly, providing an automatic resampling of the search-space. In this way, a complete “reset” of the environmental conditions is achieved, enabling the algorithm to look in areas previously considered noxious as well. Indeed, this approach is far more innovative than the Monte Carlo Filter (MCF), as no additional heuristic is required to “sense” the kidnap event since it is automatically handled by the dynamics of the equations.

3.2 Foreground Level: Multi-Hypothesis Choice and Interpretation

The competitive logistic model presented in the last section and its implementation within the BCGA represent a simple but flexible model for multi-hypothesis. Depending on the problem issues, a set of more complex equations and corresponding behaviours can be devised, as shown below in some practical examples. In the global localisation problem, it is often the case that the hypothesis choice strategy is directly related to the algorithm. Strategies might include maximum probability, maximum fitness, et cetera. In a wider context, referring to sensor fusion, the multi-hypothesis characterisation and its interpretation can be divided and independently carried out. More specifically, for any general problem setting, two possible strategies can be devised:

- Augment the complexity of species evolution model and keep a naive decision strategy.
- Keep the species evolution model simple and design a set of more accurate decision strategies using the distributions resulting from a simple BCGA.

Augmenting the model complexity requires a deeper investigation of the robot dynamics and behaviours related to the environment and the sensor measurements. Conversely, while a more simplistic model might be less robust in carrying on the multi-hypothesis, an accurate foreground strategy could compensate for this shortcoming. It is worth noting that the decision to modify the reproduction equation or the hypothesis choice depends highly on the experiment scenario. If the robot measurements are reliable, a naive reproduction scheme may be sufficient when combined with an accurate kinematic model. If the measurements are not sufficiently reliable, as in the case of laser sensors striking glass or when complex robot movements lead to phenomena such as sliding, the policy must be further investigated.

Choosing the best hypothesis is a good example of naive foreground strategy. In the case of the BCGA, the densest colony within the most nutrient area is selected. Unfortunately, this solution can lead to unrealistic optimum fluctuations. A more robust technique is achieved by introducing a weighted mobile temporal mean of the most likely hypotheses. If the aim is to preserve all plausible hypotheses, a proper multi-tracking strategy might be considered, e.g., performing a clusterisation over the colonies and describing the trajectory of each hypothesis by the barycentre of a cluster. Alternatively, the bacterial reproduction schema might be modified when sensor data are known to be unreliable.

In this paper, three different policies have been investigated:

- *Best colony* for simulation environment.
- *Mobile temporal mean* for real environment.
- *Modified reproduction* for real environment.

The mobile temporal mean has been adopted as a good compromise between efficacy and simplicity. Note that even though a similarity with the weighted resampling step of the MCF might be found when using the mobile temporal mean as a foreground strategy, a fundamental difference arises. In the MCF, weights affect the survival of hypotheses (particles), while in the BCGA, weights are introduced only to perform a comparison among colonies.

The modified reproduction is instead designed to augment the robustness of the hypotheses survival against measurement faults. The fitness reproduction chances of a bacterium do not depend only on fitness f_1 and density f_2, but also on ancestoral characteristics. The idea is that a bacterium, when reproducing, transmits its genes to the progeny and determines if they will be more or less effective in reproduction during following generations. A way to express this with formulae is:

$$r = f_{1_i}(1 - f_{2_{i+1}}) \tag{16}$$

$$h = \frac{1}{I}\sum_{k=1}^{i-1} f_{1_k} \tag{17}$$

$$\hat{r} = r + h - rh \tag{18}$$

where r is the reproduction probability previously introduced, h is the "genetic help" (equal to the average fitness of the ancestors over the generations), and $\hat{r}$ is the modified reproduction probability. This reproduction schema provides a better estimation of the hypotheses distribution, allowing for a simple foreground strategy such as the naive best-colony strategy.

3.3 Parameter Optimisation

An open problem for the MCF, the BCGA and related techniques is the parameter optimisation, such as the choice of the initial number of particles (or bacteria) or the definition of the variance for reproduction area. An *a priori* determination of these parameters is difficult. It can depend on the size of the deployment area, the ambiguity of both paths and sensors, as well as the kinematic model reliability. If the real robot path is available, the algorithm can be run several times with different parameter configurations and the resulting tracking errors can be compared through statistical tests. This way, parameters can be optimised and a satisfying performance can be achieved, lowering the number of bacteria.

In this study a non-parametric Wilcoxon rank-sum test [20] was adopted to compare median error vectors on the iteration steps. The Wilcoxon rank-sum test is a non-parametric statistical analysis of the differences in the distributions of two groups. This test is the equivalent of the Student's t-test for normal distributions, but relaxes the Gaussian requirement and allows for comparison through median and rank.

The test can be useful in performance comparisons when two robust indicators are derived from experimental settings. For this study, a performance indicator vector was generated with the aim of measuring the model behaviour over time. Specifically, at each time step, the median (or mean) of the pose error resulting from 50 independent model runs is taken. In this way, a non-parametric distribution of pose errors over time is achieved. If two models have to be compared, e.g. the BCGA against the MCF or two BCGA with different parameter settings, the corresponding performance indicator vectors are calculated and compared with the rank sum statistics. Thereby, probability values explaining the model differences are obtained.

3.4 Computational Complexity

To evaluate the ability of an algorithm to run in an on-line context, the computational complexity becomes a very useful analysis. Therefore, a detailed theoretical study has been performed along with an empirical validation of the obtained results. The algorithm presents two nested loops in which the dominant operation is the density estimation function, linear with the population size. At first glance, the complexity turns out to be $\mathcal{O}(n^3)$ in the worst case, $\mathcal{O}(n^2)$ in the best case, and $\mu \cdot n^3 = \mathcal{O}(n^3)$ in the mean case, where $\mu \in [0,1]$ is the mean reproduction factor. However, the density function can be dynamically calculated with increasing complexity, linear with the first loop, thus the worst case can be reduced by a factor of two and the mean case becomes $\mathcal{O}(n^3) = \frac{\mu \cdot n^2 \cdot (n+1)}{2}$. The complexity remains cubic, but with low constants. Coupled with the fact that in general the BCGA needs a lower number of bacteria compared to the number of particles needed for the MCF, its use in an on-line context is favourable. The next step will be an even lower bounded implementation.

4 Experiments

The proposed Bacterial Colony Growth Algorithm has been thoroughly investigated in both a simulated environment and with real robot data. Simulations were fundamental to tune the algorithm parameters and explore kidnap, while experiments carried out with the robot showed the capability of the algorithm to solve the localisation problem in different real-world contexts.

4.1 Problem Settings

Computer Simulations. Simulations have been carried out in a framework developed under Matlab by the authors. This framework provides different kinematic models for the robot, such as the unicycle model, as well as an emulation for several sensors such as a laser rangefinder. Moreover, the environment is described by a set $\mathcal{M}$ of segments. This framework supports both a complete simulated context as well a test-bed to run data coming from a real robot. These two different operative modalities turn out to be very useful, both to test the correctness and the effectiveness of the algorithm.

Real Robot Context. Experiments have been carried out on the mobile platform ATRV-Jr manufactured by iRobot. It is a skid steering vehicle mainly designed to operate in outdoor environments. The ATRV-Jr has 4 wheels differentially driven by 2 DC motors: the motion is achieved by a differential thrust on the wheel pairs at the opposite sides. The mobile robot is equipped with 17 sonar rangefinders, a laser scanner (Sick LMS-220), an inertial platform (Crossbow DMU-6X), and a GPS receiver (Garmin GPS35-HVS). The sensory system is connected to the ATRV-Jr's on board PC (Pentium II, 350 MHz) running Linux, through serial ports on a Rockeport multiserial port card. The robot is delivered with a software development environment called MOBILITY, which provides full access to the software servers available on the mobile platform. Each server is assigned to control a specific hardware component (sensors and actuators). In this way all interfaces are reachable from the network exploiting a CORBA interface.

4.2 Results

4.2.1 Simulations

The simulated environment was configured as a large indoor area with several ambiguous zones (rooms), with a few poses uniquely defined (Fig. 1). The robot was simulated moving along a fixed path for 300 steps (step interval at 1s). A kidnap condition at time $t = 100$ was added. The simulated laser sensors had a limit of 8m, while two random zero-mean artificial noise variables were added to the kinematic model and to the observation model respectively.

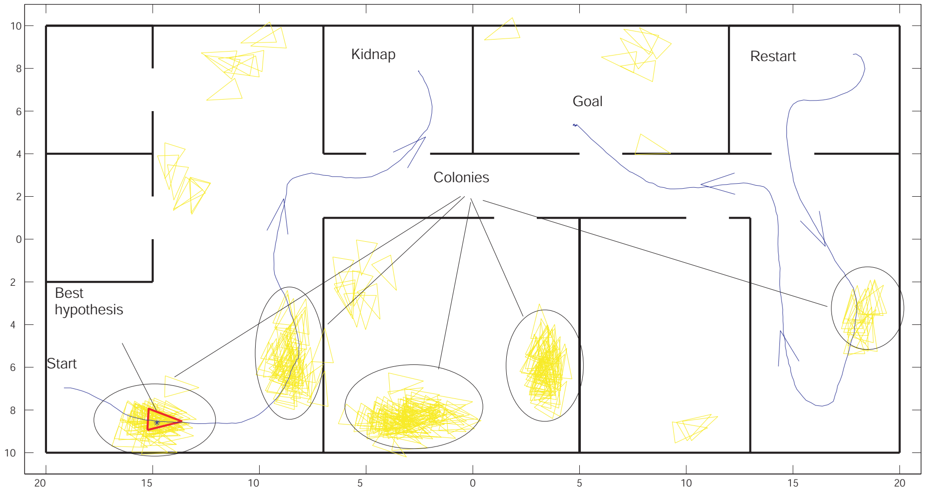

Figure 1: Simulated environment: multi-hypothesis keeping.

In order to assess the BCGA performances and robustness, a preliminary phase of parameter tuning was performed. Then, 50 independent test runs were carried out and a statistical comparison against a Monte Carlo Filter (MCF) [8] was made.

The final BCGA, after tuning, was configured with an initial random population of 300 bacteria, a maximum colony fraction size $\nu = 0.3$, a colony radius

$\sigma_r = 10m$, a sensor measure deviation of $\sigma_m = 0.1$, a tolerance in pose of $\sigma_x^2 = \sigma_y^2 = 0.1$ and $\sigma_\vartheta^2 = 0.05$. The MCF considered the same number of particles and was independently tuned on different variances.

The strategy for the best hypothesis selection was the same for both algorithms: the best particle for the MCF and the best bacterium in the densest colony for the BCGA. This is a naive strategy for the hypothesis choice (in Section 4.2.2 a set of more effective strategies are presented) but it turned out to be satisfactory in this context.

According to the simulation results, the BCGA algorithm was able to carry on the multi-hypothesis and successfully localise the robot after a few iterations. It was also able to quickly re-localise the robot when a kidnap occurred (Fig. 2). As seen by comparing Fig. 2 with Fig. 3, the median error of the BCGA is almost always lower than that of the MCF. Moreover, a non-parametric Wilcoxon rank sum test [20] showed that the BCGA significantly outperforms the MCF (ranksum=127386, z_val=17.798, $p \simeq 0$).

4.2.2 Robot in Real Environment

The ATRV-Jr was put in three indoor office environments:

- *Corridor.*
- *Lobby.*
- *Entire building floor.*

The environments were selected with increasing complexity and size. All of them contained ambiguous areas (including corridors, similar rooms, et cetera) and places in which both sensors and kinematics fail (glass doors, smooth floors, et cetera). Laser rangefinders were set to high definition and small range mode (8m), so that the overall coverage of the environment would not always be guaranteed.

The Corridor. The robot moved through the corridor, making 180° U-turns at each dead-end (Fig. 4). The sampling frequency was 5Hz and an accurate Kalman path estimation was available for comparison. The environment featured highly ambiguous pathways and areas, especially in the middle of the corridor and in the two almost identical niches at each end. Tracking was further complicated due to sliding phenomena impacting encoder data and noise affecting laser measurements (especially in the U-turn, where glass doors were also present). The *foreground* strategy was not limited to the trivial best-colony (or best-particle) choice. More complex problem settings demanded more robust hypotheses discrimination. Experimental results suggested that the simple competitive-logistic model was powerful enough to carry on the multi-hypothesis. However, a better tracking performance was obtained by exploiting the modified reproduction schema.

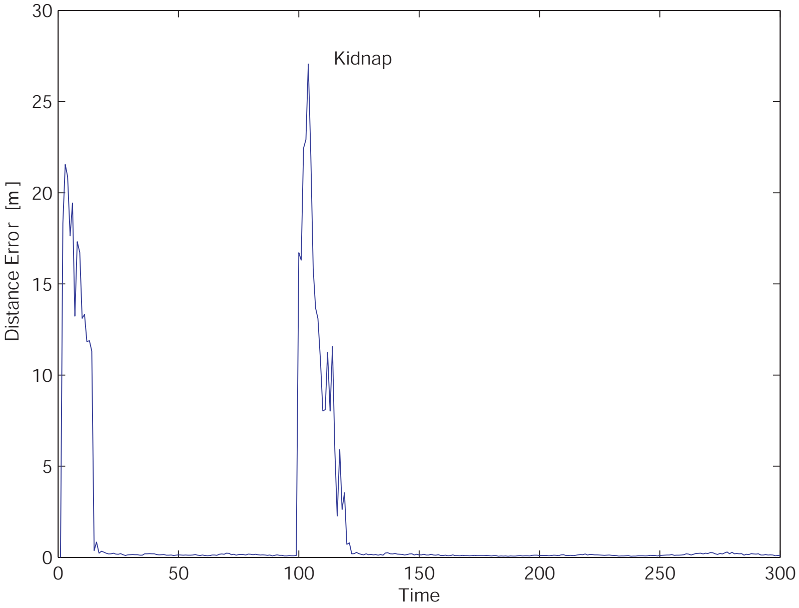

Figure 2: Simulation environment. BCGA: Median pose error over 50 trials. Naive best hypothesis choice.

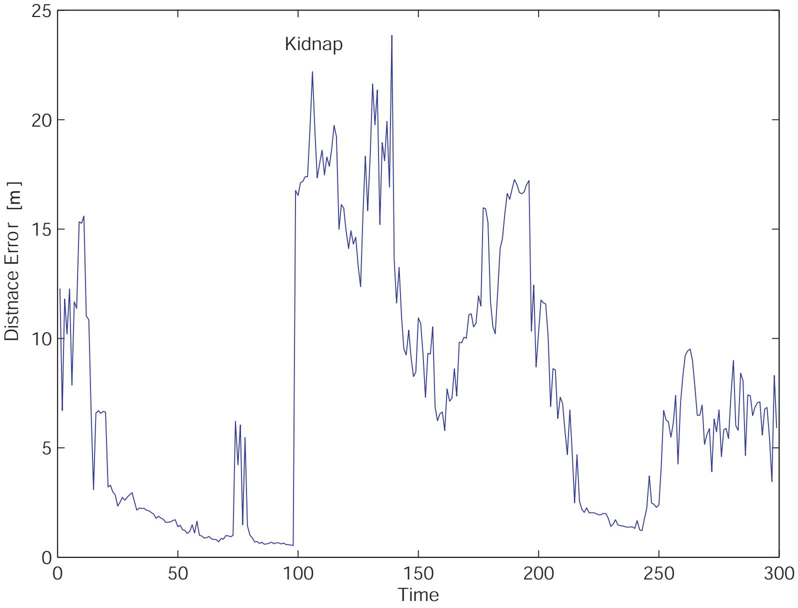

Figure 3: Simulation environment. Monte Carlo Filter (MCF): Median pose error over 50 trials. Best particle kept.

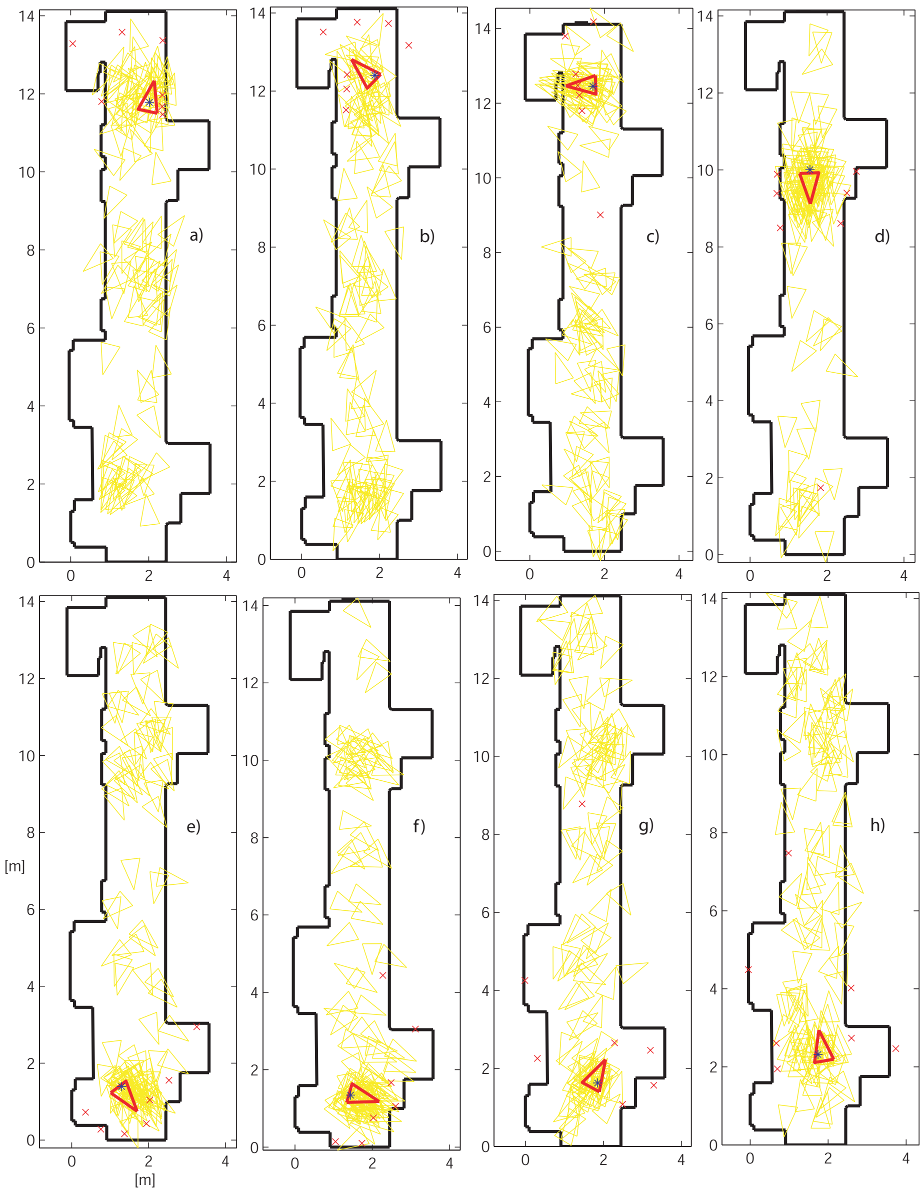

Figure 4: Real environment - Corridor - BCGA.

In particular, the following two configurations were taken into account:

1. Mobile temporal mean with the simple-competitive logistic law.
2. Modification of the reproduction law as in eq. (18).

In order to assess robustness and performance, a set of 50 independent runs were collected both for the first and the second strategies, with different parameter settings. For the mobile mean strategy, the optimally tuned BCGA was set up with the following parameters: an initial random population of 200 bacteria (as it was a smaller area compared to the simulated one); a maximum colony fraction size $\nu = 0.5$; colony radius $\sigma_r = 1m$; sensor measure deviation of $\sigma_m = 0.1$; and tolerance in pose of $\sigma_x^2 = \sigma_y^2 = 0.05$ and $\sigma_\vartheta^2 = 0.005$. Colonies grew and moved coherently with the robot poses, except in the region corresponding exactly to the U-turn. Here, the longest lasting colony (and so far, the correct one) depleted (due to the inability of sensors to properly work in presence of elements made with glass and the imprecision of the kinematic model), but recovered rapidly.

Another difficulty occurred in the middle region of the corridor. Due to the symmetry of the environment, two high-density colonies were growing and moving in opposite directions. The depletion experienced during the U-turn, along with the similarity of sensor data readings, due perhaps to the limited laser range, made the best mobile mean fail occasionally. Fig. 4 shows several steps of the algorithm's execution, where the thick (red) triangle represents the best hypothesis while the (blue) star is the real robot pose. Colonies are created w.r.t. the locations which better match with the sensor data, e.g. steps a, b, f. Colonies expanded (enhancing the state-space exploration) when in the presence of ambiguous areas, data or kinematic failures, e.g. steps g, h. Fig. 5 shows the median tracking error over 50 independent runs. Note that the mobile mean policy leads to a quick recovery from the U-turn depletion, even though problems in the middle corridor are experienced.

Experimental results indicate that the modified reproduction schema combined with the naive best-colony strategy performs better. In particular, a lower localisation error is experienced and a reduced number of bacteria is required in order to successfully localise the robot (tests were made with 30 and 300 bacteria). Although a better performance is always experienced when the number of bacteria is increased (no matter what strategy is adopted), the second strategy outperforms the first even when considering only 30 bacteria. In this context, the Wilcoxon-test could be properly exploited to find out the optimal number of bacteria to use w.r.t. a desired error level. Fig. 5 and 6 show the algorithm performance when considering the mobile mean and the modified reproduction schema with the best-colony strategy.

The Lobby. This second environment (Fig. 7) presents a wider area when compared to the corridor previously discussed. Here, the robot started from the bottom and travelled upward, turning around and returning to the bottom again. The environment was less ambiguous, but the available map was less accurate as well, e.g. the slope of the incline on the top wall was incorrect. Again, 50 experiments were run and a Kalman path estimation was available for comparison.

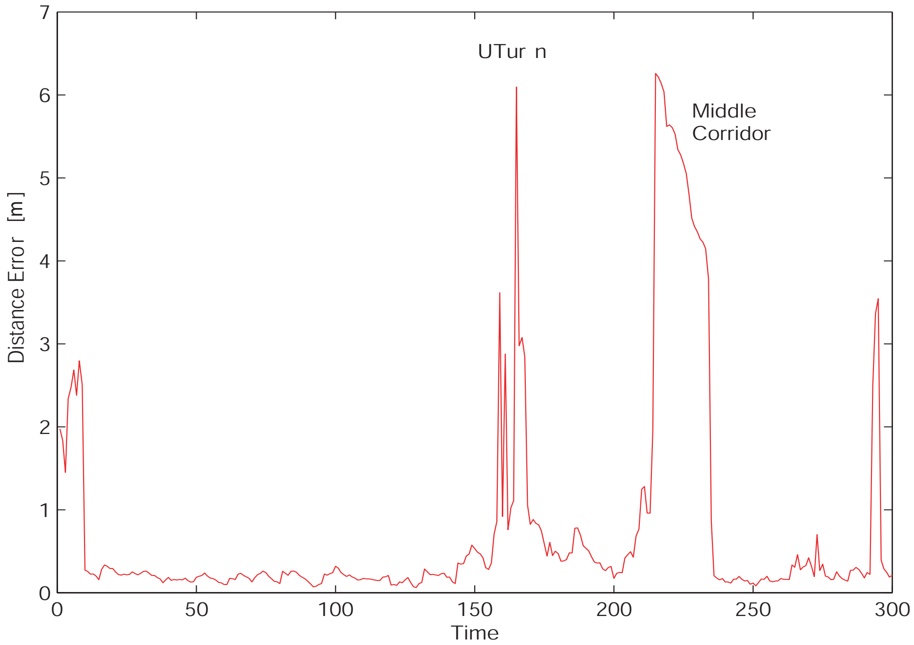

Figure 5: Real environment - Corridor - BCGA: Median pose error over 50 trials. Mobile temporal mean with the simple-competitive logistic law.

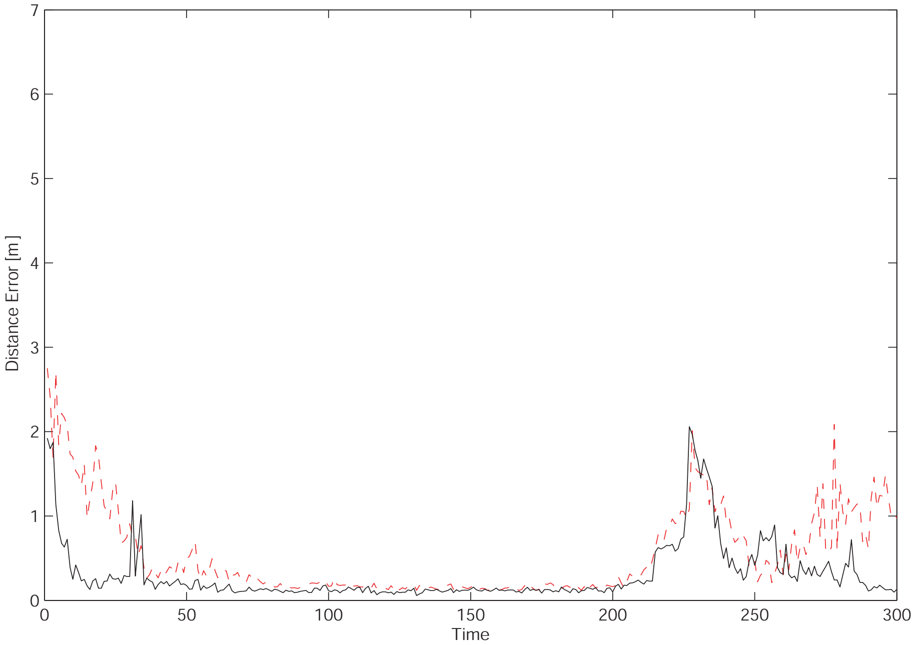

Figure 6: Real environment - Corridor - BCGA: Median pose error over 50 trials. Modified Reproduction Law with Naive best hypothesis choice. Runs with 300 (solid black line) and 30 (dash red line) bacteria (Color Online)

The modified reproduction schema presented in eq. (18) was used and the number of bacteria was varied (30, 100 and 300). The rank-sum statistics again showed better performances for the 300-bacteria (which explains the lower recovery time after measurement failures, Fig. 8 and 9). In addition, the median error was below 0.5 meters for all settings. In the turning region, although the algorithm suffered from map inaccuracy, it was robust enough to track the robot (Fig. 8 and 9). Fig. 7 shows several steps of the algorithm execution. The thick (red) triangle represents the best hypothesis, while the (blue) star is the real robot pose. As for the previous experiment, colonies were created w.r.t. the locations which better matched the sensor data, e.g. steps b, c, d. Moreover, an expansion of colonies was noticed when in the presence of environmental ambiguities, e.g. steps a, f.

Entire Building Floor. This is the largest environment where the BCGA was tested. It is the first floor of the Computer Science Engineering Dept. of Roma TRE University. It features a smooth, glossy ceramic floor, rough white walls and several glass doors and windows. The robot started from the bottom-left small niche and moved towards the first large area with the pillar surrounded by glass doors. Then, it continued through the left corridor and finally turned right into the upper horizontal corridor (Fig. 10). The total path was 1000 time steps, with sampling frequency at 5Hz. Also in this case, a reliable Kalman path estimation was available to evaluate the algorithm tracking capability.

The BCGA was run 50 times with three different population sizes (300-150-50). With less than 150 bacteria it was almost impossible to find and track the robot, while with 300 bacteria the pose error was acceptable. The modified reproduction strategy turned out to be the only one able to provide good performance, even though some problems were experienced, in particular, along the last part of the path (the long corridor). Fig. 11 shows the median errors w.r.t. a population size of respectively 300 (solid black line) and 150 (dash red line) bacteria.

5 Conclusions

This paper introduces a new, biology-inspired robot localisation approach. The framework, the *Bacterial Colony Growth Algorithm*, is composed of two different levels of execution: a *background level* and a *foreground level*. The first takes advantage of models of species reproduction to maintain the multi-hypothesis, while the second selects the best hypotheses according to an exchangeable specialised strategy, usually problem dependent. Indeed, this modular structure makes the algorithm very adaptive when considering different scenarios and objectives.

After a preliminary set-up phase, several experiments were carried out in both computer simulation and real world contexts. Simulations showed the effectiveness of the algorithm in carrying on the multi-hypothesis when in the presence of environmental ambiguities. In addition, when the tracking capabilities of the BCGA and MCF were compared, the BCGA showed better performance. This can be explained by considering the advantages of the BCGA over the MCF:

1) BCGA has a specific framework to maintain the multi-hypothesis.

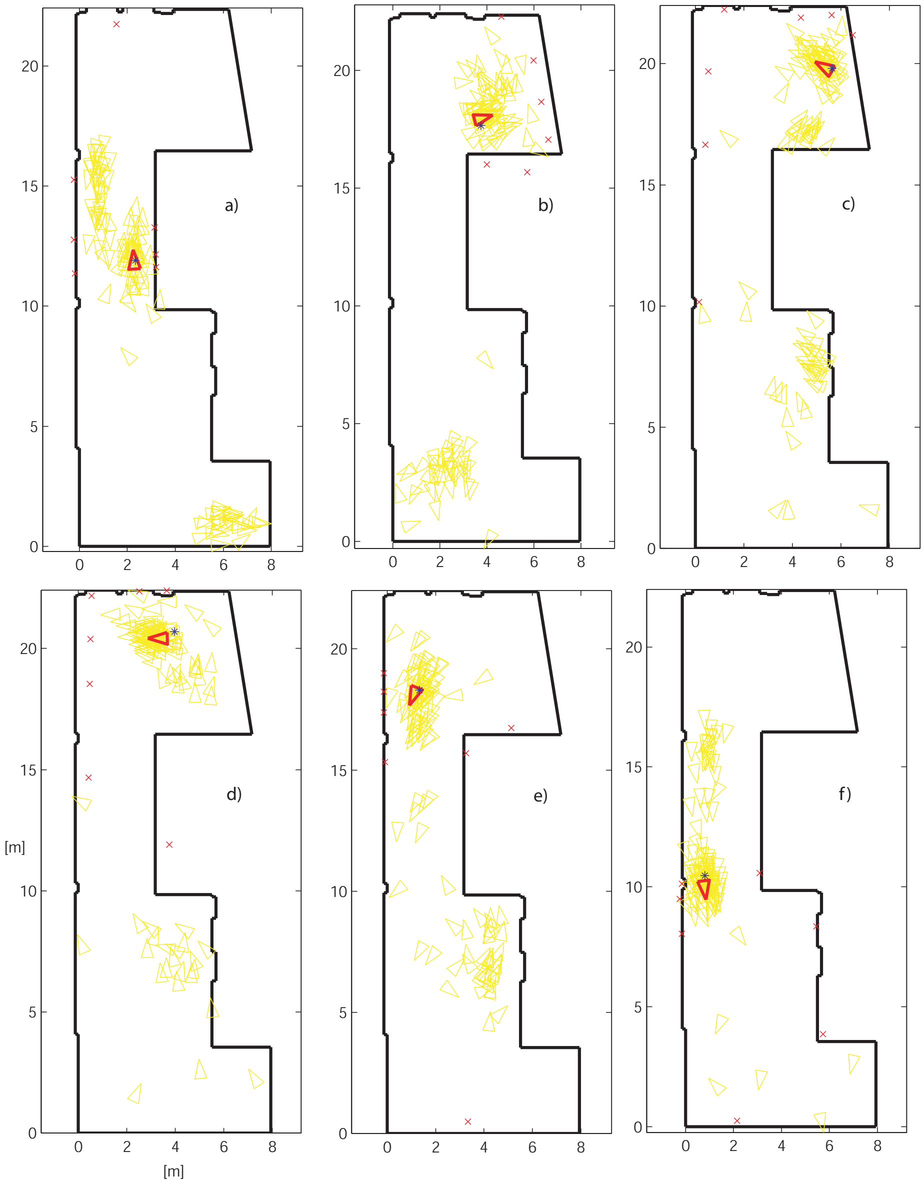

Figure 7: Real environment - Lobby - BCGA.

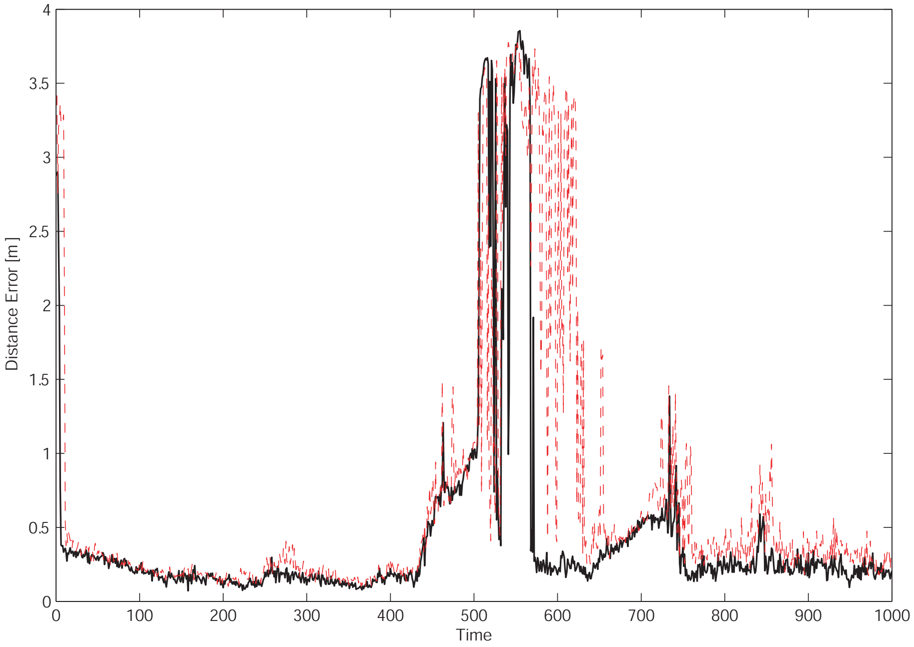

Figure 8: Real environment - Lobby - BCGA: Median pose error over 50 trials. Modified Reproduction Schema for best hypothesis choice. Runs with 300 (solid black line) and 100 (dash red line) bacteria.

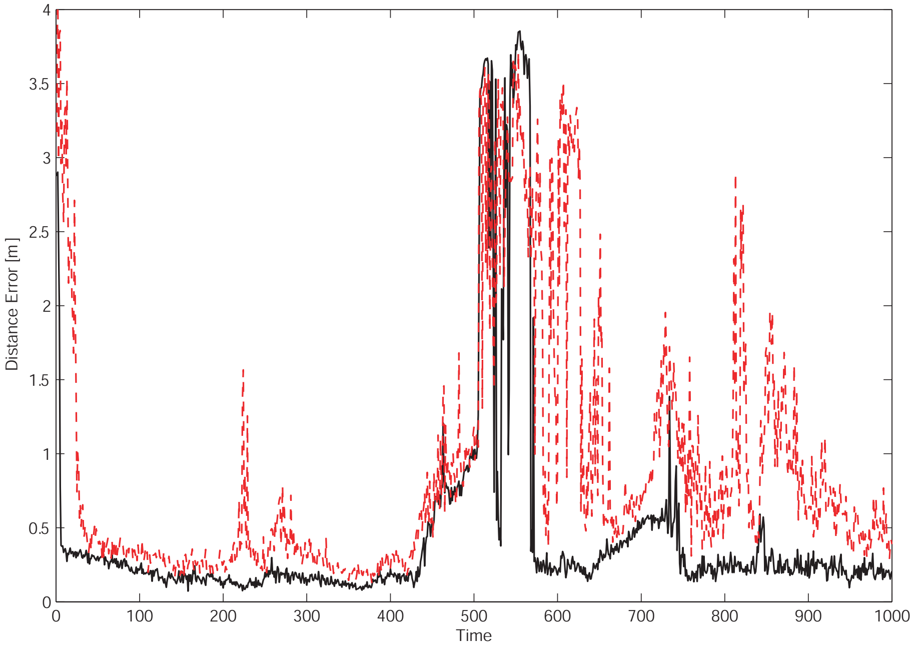

Figure 9: Real environment - Lobby - BCGA: Median pose error over 50 trials. Modified Reproduction Schema for best hypothesis choice. Runs with 300 (solid black line) and 30 (dash red line) bacteria.

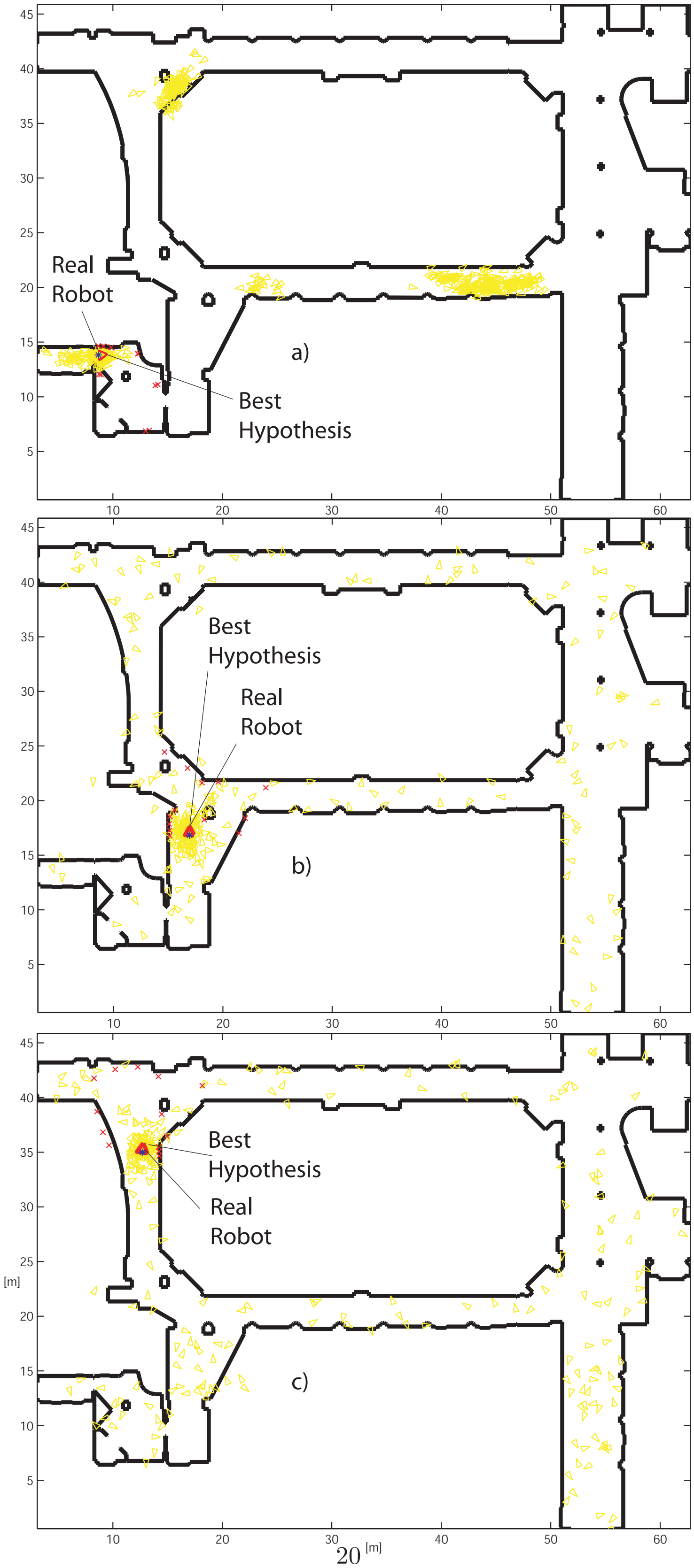

Figure 10: Real environment - Entire Building Floor - BCGA.

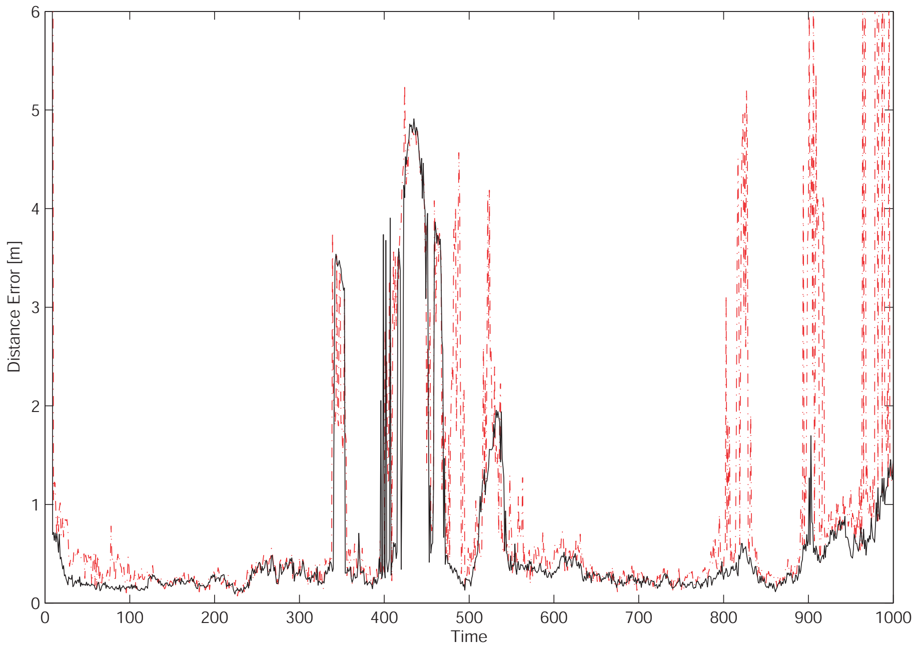

Figure 11: Real environment - Entire Building Floor - BCGA: Median pose error over 50 trials. Modified Reproduction Schema for best hypothesis choice. Runs with respectively 300 (solid black line) and 100 (dash red line) bacteria.

2) BCGA has a controlled bacteria reproduction with implicit bacterial distribution recovery.

3) BCGA has a de-coupling between hypothesis building/maintaining (background strategy) and hypothesis interpretation (foreground strategy).

The first point is a consequence of the competitive equations that form the foundation of the algorithm. In fact, the logistic model leads to concurrency and parallel survival of colonies when in the presence of more than one plausible nutrient area (i.e. possible robot path). This behaviour can be further tuned through the single-colony size parameter.

The second point is the result of the reproduction strategy. In fact, when in the presence of good nutrient conditions, bacteria tend to reproduce in a small region, forming a colony or augmenting the pre-existing colony size. Conversely, each bacterium tends to reproduce either nearby or move around if the area becomes noxious. Note that the possibility to move around is driven by a Gaussian distribution, whose standard deviation is a function of the nutritional condition of the area where the bacteria is located. This leads to a *self-adaptive* phenomenon where areas are more or less broadly explored w.r.t. the environmental condition. Obviously, if the nutrient condition is near zero, the Gaussian distribution tends towards the uniform distribution, providing an automatic re-distribution of the bacteria.

The third point underscores the most important aspect of the algorithm, which is the two-level structure. Indeed, it provides the advantage of de-coupling the search-space investigation from the solution interpretation. This is achieved by exploiting an ad-hoc exchangeable strategy (used at the foreground level to select the best solution among colonies), and the logistic equation (used at the background level to let bacteria reproduce without being affected by the best-hypothesis choice). This is the most important novelty of the framework when compared to the MCF. As opposed to the MCF where particle weights take part in the resampling step, conditioning the survival of the hypotheses, in the BCGA, the building/evolution of hypotheses is independent from their interpretation.

Finally, a performance analysis of several real-world scenarios was also carried out. Three different environments with different characteristics and incremental difficulties were exploited. Additional tracking strategies, more suitable for a real context, were devised and discussed. According to experimental results, the BGCA was shown to maintain the multi-hypothesis in these scenarios. Moreover, thanks to the specialised foreground strategies, satisfactory tracking capabilities were achieved.

Several interesting challenges still remain for future work. First, the model parameters should be estimated more accurately within a preliminary validation phase. Second, a better investigation should be performed in order to reduce the computational complexity of the framework. Finally, the model of species evolution could be further refined by introducing additional terms, e.g., flexible death-rates.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, *A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking*, IEEE Transaction on Signal Processing **50** (2002), no. 2.

[2] D.J. Austin and P. Jensfelt, *Using multiple gaussian hypotheses to represent probability distributions for mobile robot localization*, Proc. of the 2000 IEEE Int. Conference on Robotics and Automation, 2000.

[3] W. Burgard, A. Derr, D. Fox, and A. B. Cremers, *Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach*, Proc. of the International Conference on Intelligent Robot and Systems, 1998.

[4] W. Burgard, D. Fox, D. Hanning, and T. Schmidt, *Estimating the absolute position of a mobile robot using position probability grids*, Proc. of the Fourteenth National Conference on Artificial Intelligence, 1996, pp. 896–901.

[5] K. Burrage and P. M. Burrage, *Numerical methods for stochastic differential equations with applications*, SIAM, 2003.

[6] A. Doucet, *Monte carlo methods for bayesian estimation of hidden markov models. applications to radiation signals.*, Ph.D. thesis, Univ. Paris-Sud, Orsay, 1997.

[7] D. Fox, W. Burgard, and S. Thrun, *Markov localization for mobile robots in dynamic environments*, Journal of Artificial Intelligence Research **11** (1999), 391–427.

[8] N. De Freitas, A. Doucet, and N. J. Gordon, *Sequential monte carlo methods in practice*, Springer, 2001.

[9] A. Gasparri, S. Panzieri, F. Pascucci, and G. Ulivi, *A spatially structured genetic algorithm on complex networks for robot localization*, In Proc. IEEE Int. Conf. Robotics and Automation, 2007, Rome, Italy.

[10] P. Jensfelt and S. Kristensen, *Active global localization for a mobilt robot using multiple hupothesis tracking*, IEEE Transaction on Robotics and Automation (ICRA'01) **17** (2001), no. 5.

[11] R. Kalman, *A new approach to linear filtering and prediction problems*, Transactions ASME Journal of Basic Engineering **82** (1960), 35–44.

[12] J. Kennedy and R.C. Eberhart, *Particle swarm optimization*, Proceedings of the 1995 IEEE International Conference on Neural Networks, 1995, pp. 1942–1948.

[13] G. Kitagawa, *Monte carlo filter and smoother for non-gaussian nonlinear state space models*, J. Comput. Graph. Statist. **5** (1996), no. 1, 1–25.

[14] T. Malthus, *An essay on the principle of population*, London, printed for J. Johnson, in St. Paul's Church-Yard (1798).

[15] H. P. Moravec and A. Elfes, *High resolution maps from wide angle sonar*, In Proc. IEEE Int. Conf. Robotics and Automation, 1985, pp. 116–121.

[16] D. Parrott and X. Li, *Locating and tracking multiple dynamic optima by a particle swarm model using speciation*, IEEE Transactions on Evolutionary Computation, 2006, pp. 440–458.

[17] S. Schnell and T. E. Turner, *Reaction kinetics in intracellular environments with macromolecular crowding: Simulations and rate laws*, Prog Biophys Mol Biol, 2004, pp. 235–260.

[18] P. F. Verhulst, *Recherches matematiques sur la loi d'accroissement de la population*, Noveaux Memories de l'Academie Royale des Sciences et Belles-Lettres de Bruxelles **18** (1845), no. 1, 1–45.

[19] V. Volterra, *Variations and fluctuations of the number of individuals in animal species*, in Animal Ecology, Mc-Graw Hill (1931), no. 1.

[20] F. Wilcoxon, *Individual comparisons by ranking methods*, Biometrics Bulletin (1945), no. 1, 80–83.