Collective Multi-Agent Navigation Model Based on Bacterial Quorum Sensing

Edwar Jacinto Gómez¹, Mauricio Giral², Fredy Hernán Martínez Sarmiento

Fecha de recepción: 14 octubre de 2014  Fecha de aceptación: 18 de septiembre de 2015


Abstract
We present a model for the study, analysis, design and evaluation of collective multi-agent navigation for autonomous robots based on behaviors observed in bacteria. The system consists of a set of simple agents (artificial bacteria), which through readings and local interaction are self-organized to navigate along the environment. Given the parallel structure, also happens to be a very robust solution. We show the basic structure proposal for the design abstracting the characteristics of the biological model, together with an analysis of stability and convergence.

Keywords: collective navigation, intensity gradient, quorum sensing

Resumen
Se presenta un modelo para el estudio, análisis, diseño y evaluación de navegación colectiva multi-agente para robots autónomos basado en comportamientos observados en bacterias. El sistema consiste de un conjunto de agentes sencillos (bacterias artificiales), los cuales a través de lecturas e interacciones locales se auto-organizan para navegar a lo largo del ambiente. Dada la estructura paralela, también resulta ser una solución muy robusta. Se muestra la estructura básica propuesta para el diseño abstractando las características del modelo biológico, en conjunto con un análisis de estabilidad y convergencia.

Palabras clave: navegación colectiva, gradiente de intensidad, quorum sensing
INTRODUCTION

The problem of finding a robotic navigation path refers to make an agent (an object with physical dimensions) to move from one point (current configuration) to another point (desired configuration) in a navigation environment, avoiding obstacles with a minimum margin. This is currently an unsolved problem in robotics, and in fact, one of the most active fields worldwide.

The exact definition of a path for a real application is a computationally intractable problem (Idsardi, 2006). It is possible to significantly simplify the problem using geometric algorithms on the environment, such as the case of Voronoi diagrams (Xia y Xiangmin, 2014), (Kuo-Ho, Tan-Phat, Chan-Yun, y Wen-June, 2014), (Wei-Che, Chan-Yun, Kuo-Ho, y Yi-Hong, 2014), (Unghui, Sangyol, HyunChul, Vasseur, y Demonceaux, 2014), the A* search algorithm (Haifeng, Jiawei, Guifeng, y Yun, 2014), and in general, cells space decomposition algorithms (Holman, Jacinto, y Martinez, 2015), (Jan, Chi-Chia, Wei, y Ting-Hsiang, 2014), (Goyal y Nagla, 2014), (Delgado, Casallas, y Jacinto, 2014). However, these strategies can only produce useful results in real applications when the navigation environment is static, that is, it remains unchanged over a considerable period of time. This condition, in particular, is not satisfied where there are more than one agent.

Some more efficient algorithms as potential fields have serious problems of convergence and local minima (Amorim y Ventura, 2014), (Mohammadi, Rahimi, y Suratgar, 2014), (Li y Wei, 2014), which it is a big issue in a real application. One of the most efficient and robust strategies today to real problems of navigation in dynamic environments are algorithms based on local readings, formally known as Sampling Based Algorithms SBA (Van-Dung, Dongwook, Kurnianggoro, y Kang-Hyun, 2014), (Lianhang et al., 2014), (Narayanan, Vernaza, Likhachev, y LaValle, 2013). In contrast to the above mentioned methods, in which an explicit representation of the obstacles is performed in the configuration space (a configuration that describes the position of the robot, and the configuration space C, in the set of all possible configurations (LaValle, 2006)), representation that is used to build the solution, and hence the high computational cost, in the SBA it is primarily relies on a local sensing hardware aboard the agent, which provides information on possible navigation paths. This allows working with minimum and sufficient information to define the robot motion, reducing the computational load, while answers in real time allowing changes in the environment.

Under the SBA scheme, collecting and processing information is essential to the process of mapping the environment, location in the environment and path planning. The current configuration is estimated at the stage of location by sensing the configuration space (C-space), and scheduler allows to define the desired final configuration. By its nature, the sensing process is itself a random approach which has advantages in terms of being able to provide quick solutions to complex problems, but it has the disadvantage that such solutions are often suboptimal. There is no guarantee of finding an optimal navigation solution (in those cases that this exists), SBA algorithms are able to provide a solution, if there is a solution to the problem, provided there is sufficient time for the development of the algorithm (are unbounded in time). Therefore, these schemes guarantee its success (completeness) under the concept of probabilistic completeness (it is not possible to define explicitly the time required to find the solution, but if there is a solution, the algorithm will find it).

We propose a scheme of collective bio-inspired navigation based on self-organization. Self-organization has been the subject of great theoretical research, however, its practical application in solving actual problems on the contrary has not been very noticeable, or with impact (Prokopenko, 2008). Some attempts that have been documented in this respect are the works of Mange (Mange et al., 1996), (Freitas y Gilbreath, 1980), where they document some developments on FPGAs in
bio-inspired hardware that somehow involve cellular reconstruction models, replicating them to build robust applications, systems resilience to physical damage. Such developments involve two importances biological processes at the cellular level: replication and regeneration. These two processes are crucial in any bio-inspired model that tries to replicate structures at the cellular level, such as the bacterial Quorum Sensing (QS), biological model used as reference in this research.

This paper is supported on various works related to research focused on growth and development of bacteria, both from the biological point of view and the point of view of engineering (Karafyllidis, 2011), (Wiedermann, 2011), (Besozzi, Cazzaniga, Mauri, y Pescini, 2011), (Shen y Zhou, 2010), (Niu, Fan, Tan, Rao, y Li, 2010), (Taylor, Tinsley, Wang, Huang, y Showalter, 2009). From the latter field, a lot of research is seen in search and optimization processes, specifically in computational intelligence, where the growth, development and interaction between bacteria offers interesting behavioral models (Zang, He, y Ye, 2010), (Cho y Kim, 2011), (Gómez y Rodríguez, 2011), (Goni, Redondo, Arroyo, y Castellanos, 2011), (Connelly y McKinley, 2011). In particular, it comes to have a structure capable of modeling individuals, groups of individuals, and their communication processes (Martínez y Delgado, 2010).

The paper is organized as follows. In Section 2 problem formulation is discussed, Section 3 describes the control strategy. Finally, conclusion and discussion are presented in Section 4.

**PROBLEM FORMULATION**

Let \( W \subset \mathbb{R}^2 \) be the closure of a contractible open set in the plane that has a connected open interior with obstacles that represent inaccessible regions (Gonzalez, Ghaffarkhah, y Mostofi, 2014). Let \( O \) be a set of obstacles, in which each \( O \subset O \) is closed with a connected piecewise-analytic boundary that is finite in length. Furthermore, the obstacles in \( O \) are pairwise-disjoint and countably finite in number. Let \( E \subset W \) be the free space in the environment, which is the open subset of \( W \) with the obstacles removed.

Let us assume a set of \( n \) agents in this free space. The agents know the environment \( E \) in which they move from observations, using sensors. These observations allow them to build an information space \( I \). A information mapping is of the form of the equation (1).

\[
q : E \rightarrow S
\]

where \( S \) denote an observation space, constructed from sensor readings over time, i.e., through an observation history of the form of the equation (2).

\[
\sigma : [0, t] \rightarrow S
\]

The interpretation of this information space, i.e., \( I \times S \rightarrow I \), is that which allows the agent to make decisions (LaValle, 2006).

We assume the agents are able to sense the proximity of their team-mates and/or obstacles within the environment, using minimal information. The environment \( E \) is unknown to the robot. Furthermore, the robot does not even know its own position and orientation. Our goal is to design the control rules for the \( n \) robots in order to independently solve navigation tasks in a dynamic and unknown environment.

The particular equations of motion \( x = f(x) \) for each robot is unimportant in this approach. We do not explicitly control their motions and do not even attempt to measure their precise state. Instead, we rely on the fact that the robot moves in an uncontrollable way, but the trajectory satisfies the following high-level property: For any region \( r \in R \), it is assumed that the robot moves on a trajectory that causes it to strike every open interval in \( \partial r \) (the boundary of \( r \)) infinitely often, with non-zero, non-tangential velocities.

However, the robots should not move randomly in the environment, they should jointly develop a
task. To achieve this, we propose a design model based on the collective behavior of robots. This collective behavior involves a local interaction at agent level, reflecting a system-level behavior. These behaviors in conjunction with some control modes encoded in all agents, enable the coordinated navigation of the robots.

We develop an event-based system. Each robot starts with an initial control mode. During execution, the control mode may change only when receiving an sensor observation event $y$. Let $Y$ denote the set of all possible observation events for a robot in a particular system, then $y \in Y$. The control modes of robot $i$ are set during execution according to a policy. Since state feedback is not possible, information feedback is instead used. A control policy is specified as an information-feedback mapping which enables the control mode to be set according to the sensor observation history.

**METHODOLOGY**

The multi-agent model proposed is composed of a set of artificial bacteria or agents. This set of agents, and their interactions, reflect the dynamics that will navigate. All agents are identical in design. Nevertheless, to solve tasks, each of them undergoes certain behavior inside the system along the time.

The variables of this system are defined in a continuous space. However, the different agent behaviors are triggered by certain events (event-based), which implies that the appropriate analysis model is a hybrid system (the dynamic changes from one state to another when a condition is present).

The simplified model of bacterial QS describes a set of independent cells, which under the generation of extra-cellular signals produce coordinated social behaviors (Otero, Munoz, Bernández, y Fábregas, 2004). The pathogenic bacteria that carry multicellular organisms are not virulent until they reach a sufficient majority to enforce a massive attack against the immune system. When bacteria determine that their population size is enough to trigger an attack, they transform and become virulent.

**Definition 1.** A bacterium is a pair (equation (3)).

$$V = (f, p)$$  \hspace{1cm} (3)

where $f$ is a nonnegative integer ($f \in \mathbb{Z}^+$) that indicates the amount of neighboring bacteria in direct contact, and $p$ is a point in $q$-dimensional space ($p \in \mathbb{R}^q$) (Martínez S. y Delgado, 2010).

The bacterial recognition occurs in a bacterium $V_i$ when the bacterium defines its values $f$ and $p$. This definition corresponds to an extension of the cell definition in the antibody-antigen mathematical model (Tarakanov y Dasgupata, 2000).

**Definition 2.** The population density is evaluated using the distance between bacteria (equation (4)).

$$d_y = d(V_i, V_j)$$  \hspace{1cm} (4)

as the distance between bacteria $V_i$ and $V_j$, which is calculated by any appropriate norm.

**Definition 3.** A bacterial population is defined as a nonempty set of bacteria (equation (5)).

$$W_0 = \{V_1, V_2, V_3, \ldots, V_m\}$$  \hspace{1cm} (5)

with non-zero distance among bacteria (equation (6)).
The bacteria that implement the navigation task are called here Application Bacteria, AB, and are a subset of the bacterial population (figure 1).

\[ W \subseteq W_0 \]  

That is, \( W_0 \) is the set of all bacteria in the system, and \( W \) is the set of virulent bacteria in the system.

**Figure 1.** Bacterial population and application bacteria in the proposed system.

**Source:** Own work

**Definition 4.** The neighborhood threshold \( h \), indicates the maximum amount of direct neighboring bacteria that a cell can have.

\[ d_{ij} \neq 0, \quad \forall i, j \neq i \]  

Throughout reproduction, the population size (number of agents) is always compared to \( T \), the quorum threshold, it is the parameter defining whether or not it has reached the quorum. Regarding the physical dimensions of the bacterial population, the density threshold \( h \) indicates the minimum distance needed between bacteria; in other words, the minimum distance that must exist between any pair of bacteria.

The behaviors of bacteria (self-organization) are coordinated by the following rules (the model does not include cell death):

**Reproduction Rule:** If the bacterium \( V_i \in W_0 \setminus W \) can reproduce (equation (8)).

\[ f_i < \eta \quad \text{and} \quad d_{ij} < h, \quad \forall V_j \in W \]  

then \( V_i \) must reproduce by duplicating their DNA (code) in the available medium.

In other words, if the bacterial population is very small and the non-virulent bacterium \( V_i \) can reproduce, then \( V_i \) must reproduce (must copy its code to a neighboring inactive).

**Virulence, Activation or Cell Differentiation Rule:** If the bacterium \( V_k \in W \) is the nearest to the bacterium \( V_i \in W_0 \setminus W \) among all AB (figure 1).

\[ d_{ik} < d_{ij}, \quad \forall V_j \in W, k \neq j \quad \text{and} \quad d_{ij} < h, \quad \forall V_j \in W, k \neq j \]  

and the number of bacteria in \( W \) is less than \( T \), then \( V_i \) must be added to AB (the bacterium becomes virulent).

In other words, if the AB is very small and the non-virulent bacterium \( V_i \) is the nearest to a virulent bacterium, then \( V_i \) must become virulent (must solve the task with the other bacteria of AB).

**Stability, convergence and parameters selection**

The stability and convergence of the model is dependent on the parameters selected for the design of the navigation environment. Because the schema is supported in local readings, its use in real applications requires modification (design) of the...
navigation environment. The parameters selection is essential for this design and modification of the environment.

The design of the navigation environment consists of defining the areas of \( E \) to be the final destination for the agents, and then place in them and in their neighborhoods, special marks (landmarks) that indicate this condition. Generally these landmarks are not equal, each value indicates the intensity corresponding to its position in \( E \). The intensity value is designed for the entire navigation environment, so it grows near the final destination area.

Thus, during the search process (navigation), agents interpret local readings and determine the intensity value at each point of the environment. Climbing the slope of these values, each agent follows a heuristic that allows it to choose the optimal choice in each advance as does a greedy algorithm. Although such an approach can be inappropriate for some computational tasks (the strategy stagnates in local solutions without finding the global optimal solution), for the proposed algorithm based on QS this constitutes an exploration strategy that ultimately ensures convergence in the global optimal solution.

The effect of QS on the population is to accelerate the convergence in the areas of high performance. For this, the model takes into account the population size in the area, increasing the attraction when a threshold is exceeded. The value of the slope is designed as a function of the density threshold.

The following sections will show how to model the navigation field intensity as a gradient on the environment. This section focuses only in the vicinity of an area of high performance (area to be interpreted by the agents as of interest), and how to design a simple intensity field according to a slope, considering the issues of convergence.

The gradient in the area can be modeled by a system of two autonomous first-order ODEs (Ordinary Differential Equations), model in which the slope at each point represents the intensity of the gradient. To plot the slope field, the two autonomous ODEs must be written in the form (equation (10)).

\[
\begin{align*}
\frac{dx}{dt} &= G(x, y) \\
\frac{dy}{dt} &= F(x, y)
\end{align*}
\]

That is, the first expression is the derivative of the variable represented on the horizontal axis of the environment, and the second expression is the derivative of the variable represented on the vertical axis of the environment. The origin corresponds to the center of the area of high performance.

The two functions must confine the agents at the origin. Since the slope of the two curves are who push the agents, it is possible to think of a curve design for \( x \) and \( y \) as shown in figure 2.

![Figure 2. Linear approximation of intensity gradient behavior in an area of high performance.](source: own work)

These two curves are designed in such a way that with their growth there is no possibility of way between them. Curve \( y \) (green) is a reflection of the curve \( x \) (blue), and the slopes push towards the origin. To facilitate the calculation of derivatives, these curves can be replaced by exponential approximations (equation (11)).
In these two curves is easy to define the behavior: \( a \) is the reference value and \( t \) is the time constant, which directly affects the slope of the curve, and can be defined as a function of the density threshold.

To represent graphically the slope field is necessary to calculate the derivatives of the curves (equation (12)).

\[
\frac{dx}{dt} = -\frac{a}{\tau \times e^{\frac{y}{\tau}}}, \text{and} \frac{dy}{dt} = \frac{a}{\tau \times e^{\frac{y}{\tau}}}
\] (12)

Since these curves are exponential, we subtract to each one the value of the other variable to prevent grow up indefinitely (y to x and x to y). The figure 2 shows the slope field for a time constant of: \( 5\tau = 5 \Rightarrow \tau = 1 \).

From the model and these behaviors is possible to conclude some general characteristics in terms of stability and convergence.

**Lemma 1:** Let \( p = \{p_1, p_2, \cdots, p_m\} \) be a set of points drawn randomly in \( E \) on a two-dimensional Euclidean plane \( \mathbb{R}^2 \), each with an associated slope value \( m \) according to some law of navigation. Let \( G(x, y) \) be a function on \( \mathbb{R}^2 \) that assigns a slope value to each point \( p \in \mathbb{R}^2 \), such that the maximum value is assigned to the central point of the area of high performance. Then there exists a point \( p_0 \) such that \( G(p_0) = m_0 \geq G(p) \ \forall p \in \mathbb{R}^2 \) where \( p_0 \) is the closest point to the center of the area of high performance.

**Proof:** By construction the maximum value of \( m \) is assigned to the central point of the area of high performance. If this construction is correct, \( p_0 \) always exists. \( p_0 \) is the \( p \in \mathbb{R}^2 \) closest point to the area of high performance. Furthermore, if the heuristic climbs always looking for the right next value, the agent eventually will find this point \( p_0 \) if there are no other areas of high performance.
Lemma 2: If there is at least one area solution in $E$, incorporation of QS in finding solutions does not produce stability problems in the search algorithm. By contrast, this reduces the convergence time.

Proof: The proposed algorithm based on QS alters the behavior of the function $G(x,y)$ at points near to the high performance area. This alteration is conditioned to the density threshold, producing localized alterations if the population in the area exceeds that threshold. QS increases the weight of the solution, and therefore reducing the total search time. However, QS does not change the global behavior of $G(x,y)$, therefore does not affect the convergence of the search.

CONCLUSIONS

In this paper we propose a model for collective multi-agent navigation that enables the analysis and design of navigation tasks for an autonomous group of robots. The model is inspired by bacterial interaction, and describes a simplified behavior of bacterial QS, which serves to define an artificial bacterium, its characteristics of interaction, and its basic navigation conditions. The structure assumed in the proposed model is hybrid, in which behaviors that can be analyzed continuously are triggered by events (local readings) characterizing discrete transition. Taking as a design strategy landmarks allocation based on density gradients, we demonstrated that the system is stable and converges to a solution if it exists. Thus, we show how a group of robots can be used to solve simple navigation tasks without requiring system identification, geometric map building, localization, or state estimation, using a simple minimalist design in hardware, software and algorithm operation. The interaction between the robots responds to the simplified algorithm of local communication, which guarantees a minimum sensing, what makes the strategy promissory for exploration in collapsed and unknown environments. The capabilities and system functions can be scaled changing the number of robots.

FINANCING

This work was supported by the District University Francisco José de Caldas, in part through CIDC, and partly by the Technological Faculty. The views expressed in this paper are not necessarily endorsed by District University. The authors thank the research groups DIGITI and ARMOS for the evaluation carried out on prototypes of ideas and strategies.

REFERENCES

Collective multi-agent navigation model based on bacterial Quorum Sensing

Jacinto, E., Giral, M., & Martín, F.

and demonstration. En Advanced automation for space missions, 1980 nasa/asee summer study.


Haifeng, W., Jiawei, Z., Guifeng, Z., y Yun, L. (2014). Has: Hierarchical a-star algorithm for big map navigation in special areas. En 2014 5th international conference on digital home (icdh) (p. 222-225).


Collective multi-agent navigation model based on bacterial Quorum Sensing

Jacinto, E., Giral, M., & Martín, F.


