

A biologically inspired controller to solve the coverage problem in robotics

Iñaki Rañó

Iñaki Rañó is with the Computer Science Research Institute,
Intelligent Systems Research Centre, Ulster University, UK.
`i.rano@ulster.ac.uk`

José A. Santos

José A. Santos is with the Computer Science Research Institute,
Smart Environments Research Group, Ulster University, UK.
`ja.santos@ulster.ac.uk`

Abstract. The coverage problem consists on computing a path or trajectory for a robot to pass over all the points in some free area and has applications ranging from floor cleaning to demining. Coverage is solved as a planning problem – providing theoretical validation of the solution – or through heuristic techniques which rely on experimental validation. Through a combination of theoretical results and simulations, this paper presents a novel solution to the coverage problem that exploits the chaotic behaviour of a simple biologically inspired motion controller, the Braitenberg vehicle 2b. Although chaos has been used for coverage, our approach has much less restrictive assumptions about the environment and can be implemented using on-board sensors. First, we prove theoretically that this vehicle – a well known model of animal tropotaxis – behaves as a charge in an electro-magnetic field. The motion equations can be reduced to a Hamiltonian system, and, therefore the vehicle follows quasi-periodic or chaotic trajectories, which pass arbitrarily close to any point in the work-space, i.e. it solves the coverage problem. Secondly, through a set of extensive simulations, we show that the trajectories cover regions of bounded workspaces, and full coverage is achieved when the perceptual range of the vehicle is short. We compare the performance of this new approach with different types of random motion controllers in the same bounded environments.

1. Introduction

Research on the mechanisms generating animal movement has led to interesting application in different areas of robotics. Good examples are Central Pattern Generators (CPGs) (Hooper, 2001), movement primitives (Giszter, Mussa-Ivaldi, & Bizzi, 1993), and insect navigation strategies (Fraenkel & Gunn, 1961). CPGs, for example, help simplifying the control of animal walking and crawling (Ijspeert, 2008). Different frameworks of movement primitives (Ijspeert, Nakanishi, Hoffmann, Pastor, & Schaal, 2013) (Nori & Frezza, 2005) (Paraschos, Daniel, Peters, & Neumann, 2013) contribute to solve the movement planning problem in robotic manipulators and systems with many degrees of freedom (Schaal, 2006). Principles and models of sensor based animal navigation (Braitenberg, 1984) (Raño, 2014b) have been extensively used to implement biologically inspired mobile robot steering, target reaching, and obstacle avoidance (Webb, 2001) (Lilienthal & Duckett, 2004) (Salumäe, Raño, Akanyeti, & Kruusmaa, 2012) (Lebastard, Boyer, Chevallereau, & Servagent, 2012) (Yang, Patel, & Moallem, 2006). The present work proposes to solve the problem of area coverage in robotics using one such navigation strategy.

How animals navigate in the presence of stimuli has been a subject of research for many years (Fraenkel & Gunn, 1961), yet the mechanisms generating these behaviours are not fully understood even for very simple animals. There are two main strategies that animals use to control their movement towards or away stimulus sources: klinotaxis and tropotaxis. While klinotaxis exploits spatio-temporal information of the stimulus, tropotaxis relies only on spatial information, and, therefore, requires the stimulus to be sampled simultaneously at different points of the environment. A fundamental principle of tropotaxis is that animals use the bilateral symmetry of their sensing organs to sample

stimulus for senses like sight (Qian, 1997), hearing (Harper & McAlpine, 2004), and smell. Recently, for instance, it has been found that exploitation of symmetric olfactory organs occurs in larvae (Ohashi, Morimoto, Suzuki, Miyakawa, & Aonishi, 2014), rats (Rajan, Clement, & Bhalla, 2006), and even humans (Porter et al., 2007). The internal neural wiring between the perceived stimulus and the motor effectors defines the behaviour of the animal, generating positive or negative taxis (movement towards or away from a stimulus), as captured in the seminal work of Valentino Braitenberg (Braitenberg, 1984), where taxis are modelled through simple vehicles. Braitenberg vehicles have been used in robotics for several decades from their most basic formulation, to extensions with fuzzy controllers or neural networks. Therefore, they are widely used to implement bio-inspired robotic behaviours, specially when the motion relies on unconventional sensors, i.e. sensors not providing distance readings. We will show that one of these taxis models, Braitenberg vehicle 2b, can be used to solve the coverage problem, a problem often tackled as a planning problem.

The coverage problem for a single robot consist of generating a path (or trajectory) that passes through all the points in an area (or volume) while avoiding obstacles (Choset, 2001). This has multiple practical applications like floor cleaning, surface painting, demining, harvesting, and lawn mowing. Multiple solutions to the coverage problem exist, working under different assumptions about the environment to cover and the motion capabilities of the robots. These solutions can be classified along two main dimensions, one of which is on-line vs. off-line. Off-line solutions assume a known static environment and typically rely on planning and some type of cell decomposition (Latombe, 1991). The cell decomposition can, in turn, be exact or approximate, covering the whole free space or having cells with shared free and occupied space. On-line (sensor based) techniques rely on the robot’s on-board sensors to scan the environment, hence, they work in *a priori* unknown static environments. The other way of classifying coverage methods is according to whether they are complete or heuristic. Complete techniques ensure theoretically that the whole free space is visited by the robot in a bounded time, while heuristic methods mainly rely on experimental results. Because of their soundness, plenty of research efforts have focused on complete techniques, either on-line (Acar, Choset, & Lee, 2006) (Wong & MacDonald, 2003) or off-line (Acar, Choset, Rizzi, Atkar, & Hull, 2002) (Oksanen & Visala, 2009). Despite their success these approaches have some drawbacks. For instance, since these techniques were originally designed for two-dimensional flat environments, their extension to surfaces embedded in three-dimensional spaces is not straightforward (Hameed, Cour-Harbo, & Osen, 2016). Ignoring practical issues in the implementation of coverage (e.g. battery life), another important limitation of complete techniques is the need for an accurate map of the environment or (for on-line techniques) the capacity to access accurate positional information of the robot and the obstacles in the environment (Shnaps & Rimon, 2016). Finally, path planning based coverage typically involves solving NP-hard problems like the Art Gallery Problem and the Travelling Salesman Problem (Bircher et al., 2016). In summary, these techniques are not well suited for noisy or uncertain

environments, nor for computationally cheap implementations of coverage.

Heuristic techniques, on the other hand, are typically based on behaviour sequencing and different types of (biased) random motion, providing computationally inexpensive solutions with excellent experimental results (Doty & Harrison, 1993) (MacKenzie & Balch, 1996) (Schroeder, Ramakrishnan, Kumar, & Trease, 2017). Sometimes they also provide some theoretical convergence guarantees. If, for instance, the environment is decomposed into cells of the size of the robot, random walks ensure coverage in an average number of steps polynomial with the size of the environment (Lovász, 1993). However, random walk based techniques are considered to be inefficient for two main reasons; first, they ensure coverage only in the limit of time going to infinity, and, second, specific areas of the environment can be revisited multiple times, while others are not yet visited. Although it can be an advantage for applications like surveillance, the total unpredictability of the trajectories can be also seen as a limitation of these techniques. Beyond random movement, chaos has proven to be a good alternative to solve the coverage problem because of its interesting properties, like the fact that chaotic trajectories are dense and predictable in the short term. This means that the trajectory of a chaotically behaving robot will pass arbitrarily close to any point in the state space, and in a large time-scale will look like random. Chaos based approaches to coverage can be considered to fall in the middle of the spectrum between complete and heuristic methods, since chaotic trajectories are dense (complete) in some region (heuristic) of the state space. This was exploited in (Nakamura & Sekiguchi, 2001), where the Arnold equation, a chaotic system found in fluid dynamics, was used to control a wheeled robot to cover a two-dimensional workspace. In this first application of chaos to robotics, the authors demonstrate that their chaotic controller outperforms random movement based control for coverage. Like in the case of complete techniques, one big limitation of this, and other chaos based coverage approaches (see (Li, Song, Wang, Wang, & Li, 2016) and references therein), is the requirement of accurate position information of the robot. Moreover, an even more important drawback is that the existing approaches to chaotically generated coverage trajectories cannot cope with obstacles, i.e. they require special solutions to avoid obstacles. A different approach was used in (Volos, Kyprianidis, & Stouboulos, 2012) for a patrolling task, where a chaotic system was used to generate random points for a planner. They achieve fast coverage, but in this case the chaotic system is just used as a random number generator.

The Braitenberg vehicle 2b represents an extremely simple model of animal tropotaxis, which can be implemented using on-board proximity sensors (i.e. it does not require position information), or through a scalar function of the environment (if position information is available). The closed-loop dynamical system describing the movement of such a robot shows interesting properties like continuous collision free motion and sometimes chaotic trajectories. Besides the implications this has for biology, this Braitenberg vehicle can be used to solve the coverage problem using a chaos-based approach which does not require accurate position information and can handle obstacles naturally. Therefore, the main contribution of this paper is twofold. First we

present a theoretical proof that the closed-loop system describing the movement of a Braitenberg vehicle 2b is equivalent to a charge in an electro-magnetic field, and, under some assumptions, to a Hamiltonian system, i.e a charge or mass in a potential well. Therefore, the second contribution is showing that these vehicles' simple controller can be used for coverage tasks since the solutions trajectories can be chaotic, i.e. dense in the workspace. We compare the coverage performance of the chaotic trajectories with several random motion generation mechanisms. The rest of the paper is organised as follows. Section 2 reviews the quantitative model of the vehicle and presents some mathematical properties of their trajectories, the most relevant being that – under some conditions – the closed loop equations can be derived from a Hamiltonian function. Section 3 presents extensive simulations of the vehicle in different environments analysing experimentally the necessary and sufficient conditions for the system to cover the workspace. A comparison with other heuristic techniques for coverage is presented in section 4. The paper ends by drawing some conclusions and proposing ideas for further work in Section 5.

2. Braitenberg vehicle 2b as a Hamiltonian System

Among the different models of tropotaxis presented in (Braitenberg, 1984), this paper focuses on vehicle 2b, which consists of a dual-drive wheeled vehicle with two sensors symmetrically arranged at the front as shown in Figure 1. The vehicle is immersed in a scalar stimulus field which can be measured by the sensors at their respective locations. The connection between the sensors and the wheels is contralateral and excitatory as represented in the figure by the crossed lines and the '+' sign. Since the connection between the sensors and the motors is increasing, the more intense the stimulus gets the faster the wheel on the opposite side turns. The overall effect is making the robot turn towards high values of the stimulus and increase its speed, i.e. the more intense the stimulus gets, the higher the vehicle velocity. Therefore this vehicle represents a qualitative model of positive tropotaxis.

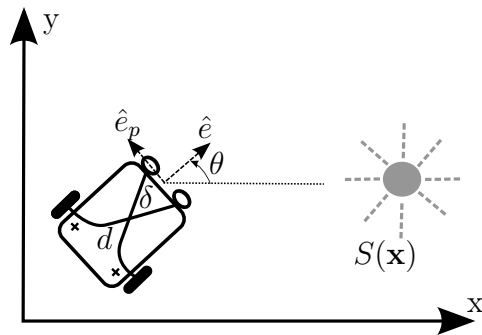


Figure 1. Internal structure of the Braitenberg vehicle 2b.

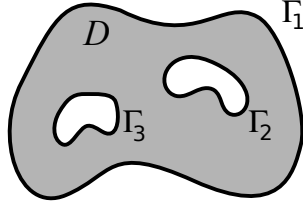


Figure 2. Domain where the stimulus is defined. The connected domain can be defined as the area enclosed by a set of curves Γ_i , one of which is the environment boundary.

2.1. Theoretical background of Braitenberg vehicle 2b

This section states the assumptions underlying the modelling of Braitenberg vehicle 2b in this work (see (Rañó, 2012) for a complete derivation of the equations). First, we can assume that the scalar function defining the stimulus $S(\mathbf{x})$ exists in a connected and compact (closed and bounded) set (see figure 2) $D \cup \partial D \in \mathbb{R}^2$, where D represents the interior of the region and ∂D its boundary. We can select ∂D such that it limits the influence of the stimulus (Γ_1 in figure 2) and defines the boundary of the obstacles in the workspace (Γ_2 and Γ_3 in figure 2), and without loss of generality we set $S(\mathbf{x}) = 0$ for all $\mathbf{x} \in \partial D$. The stimulus will be positive in the interior $S(\mathbf{x}) > 0$ for all $\mathbf{x} \in D$, and we will assume $S(\mathbf{x})$ is at least of class C^2 . We will model the connection between the sensors and the motors as a simple functional relation $F(s) : \mathbb{R}^+ \cup \{0\} \rightarrow \mathbb{R}^+ \cup \{0\}$. As already stated, vehicle 2b has an increasing connection which can be modelled constraining $F(s)$ to have a strictly positive derivative, i.e. $F'(s) > 0 \forall s \in \mathbb{R}^+ \cup \{0\}$. Since $F(s)$ represents the turning velocity of each wheel and its image is $\mathbb{R}^+ \cup \{0\}$, the vehicle does not move backwards, and we will further assume that $F(0) = 0$, i.e. the vehicle stops at the environment boundary. We will denote the Cartesian coordinates of the vehicle as $\mathbf{x} = [x, y]$ and its heading as θ , i.e. the state of the vehicle is $[\mathbf{x}, \theta] = [x, y, \theta]$, and belongs to $D \cup \partial D \times S^1$, where S^1 represents the unit circle.

The state of the differential drive vehicle 2b immersed in $S(\mathbf{x})$ and with a wheel velocity control $F(s)$ evolves according to the following system of differential equations (see (Rañó, 2012)):

$$\begin{aligned} \dot{x} &= F(S(\mathbf{x})) \cos \theta \\ \dot{y} &= F(S(\mathbf{x})) \sin \theta \\ \dot{\theta} &= \frac{\delta}{d} \nabla F(S(\mathbf{x})) \cdot \hat{e}_p \end{aligned} \tag{1}$$

where $\nabla F(S(\mathbf{x}))$ is the gradient of the composite function, $\hat{e}_p^T = [-\sin \theta, \cos \theta]$ is a unit vector orthogonal to the vehicle's head direction, δ is the distance between the sensors, and d is the wheelbase of the vehicle (see figure 1). To simplify the notation we will write $F(\mathbf{x})$ instead of $F(S(\mathbf{x}))$. Interesting properties of this dynamical system (closed-loop model) describing the movement of the vehicle can be obtained for the

working assumptions. For instance, it can be shown that the movement – with initial conditions in D – is restricted to D and never stops (Rañó, 2014a), and that – assuming the gradient does not vanish at the boundary – all points in ∂D are equilibrium saddle points. This means the movement of the vehicle will be restricted to the workspace by the boundary (Γ_1), while avoiding obstacles (Γ_2 and Γ_3).

From the equation of the linear velocity of the vehicle $v = F(\mathbf{x})$, we can compute the derivative w.r.t. time, and extend the closed-loop motion equations (1) with the differential equation governing the linear acceleration $\dot{v} = \nabla F(\mathbf{x}) \cdot \dot{\mathbf{x}}$. Therefore the movement of the vehicle is described by the following set of differential equations:

$$\begin{aligned}\dot{\mathbf{x}} &= F(\mathbf{x})\hat{e} \\ \dot{\theta} &= \frac{\delta}{d}\nabla F(\mathbf{x}) \cdot \hat{e}_p \\ \dot{v} &= v\nabla F(\mathbf{x}) \cdot \hat{e}\end{aligned}\tag{2}$$

where $\hat{e}^T = [\cos \theta, \sin \theta]$ is the unit vector in the heading direction of the vehicle. Although the position of the vehicle is represented in Cartesian coordinates, the evolution of its velocity is encoded as the vector modulus (v) and its direction (θ , which corresponds to the heading direction of the vehicle). We will denote the full state vector in this ‘mixed’ coordinate system as $\mathbf{X}_P = [x, y, \theta, v]$. If we denote $\mathbf{X}_C = [x, y, v_x, v_y]$ the state in Cartesian coordinates, we can convert the system equations (2) to Cartesian coordinates using the change of variables $\mathbf{X}_P = G(\mathbf{X}_C)$:

$$\begin{aligned}\theta &= \text{atan}\left[\frac{v_y}{v_x}\right] \\ v &= \sqrt{v_x^2 + v_y^2}\end{aligned}\tag{3}$$

where the position variables are the same for \mathbf{X}_P and \mathbf{X}_C . Now the system of differential equations (2) can be converted using $\dot{\mathbf{X}}_C = \left[\frac{\partial G}{\partial \mathbf{x}_C}\right]^{-1} F(G(\mathbf{x}_C))$, and we obtain that the motion equations in Cartesian coordinates (after arrangement of the terms) can be stated as:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= F(\mathbf{x})\nabla F(\mathbf{x}) + \left[1 - \frac{\delta}{d}\right] \mathbf{v} \times [\hat{e} \times \nabla F(\mathbf{x})]\end{aligned}\tag{4}$$

where $\mathbf{v} = [v_x, v_y]$, $\nabla F(\mathbf{x})$ is the gradient of the composite function $F(S(\mathbf{x}))$, ‘ \times ’ represents the cross product of the vectors, and we used the fact that $v = F(\mathbf{x})$. Equations (4) correspond to the Lorentz force applied to a unit charge in an electrostatic field, and a magnetic field \mathbf{B} dependent on the motion direction of the particle and the gradient of the function $F(\mathbf{x})$, i.e. $\mathbf{B} \propto \hat{e} \times \nabla F(\mathbf{x})$. If we further assume the distance between the sensors is the same as the wheelbase, i.e. $\delta = d$, the term corresponding to the magnetic force vanishes and the equations simplify to:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{v} \\ \dot{\mathbf{v}} &= F(\mathbf{x})\nabla F(\mathbf{x})\end{aligned}\tag{5}$$

As we will see, this reformulation of the closed-loop equations of Braitenberg vehicle 2b allows us to show that the dynamical system describing the movement of the vehicle fall into the type of Hamiltonian systems, and, therefore, known theoretical results from these systems can be applied to Braitenberg vehicle 2b.

2.2. Braitenberg vehicles as Hamiltonian Systems

Hamiltonian mechanics reformulates classical mechanics using the variational principle of least action, but Hamiltonian systems in general are a special type of dynamical systems with well know properties. They have an even-dimensional state space, composed by an n degrees of freedom vector of generalised coordinates \mathbf{q} , and their corresponding generalised momenta \mathbf{p} . Given the manifold in which \mathbf{q} and \mathbf{p} live, a function of \mathbf{q} and \mathbf{p} can be found, the Hamiltonian function $H(\mathbf{q}, \mathbf{p})$, which is conserved along any trajectory of the system. Moreover, the system's dynamics can be obtained from the Hamiltonian function as:

$$\begin{aligned}\dot{q}_i &= \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial p_i} \\ \dot{p}_i &= - \frac{\partial H(\mathbf{q}, \mathbf{p})}{\partial q_i}\end{aligned}\tag{6}$$

where p_i and q_i are the components of the generalised coordinates and momentum. In the case of a particle of mass m in a potential $U(\mathbf{q})$, $H(\mathbf{q}, \mathbf{p})$ takes the form $H(\mathbf{q}, \mathbf{p}) = \frac{\mathbf{p}^2}{2m} + U(\mathbf{q})$ which represents the sum of the kinetic and potential energies. This leads to the following dynamic equations:

$$\begin{aligned}\dot{\mathbf{q}} &= \frac{\mathbf{p}}{m} \\ \dot{\mathbf{p}} &= - \nabla_{\mathbf{q}} U(\mathbf{q})\end{aligned}\tag{7}$$

where the gradient $\nabla_{\mathbf{q}} U(\mathbf{q})$ is computed w.r.t. the generalised coordinates. Because the total energy is constant along a trajectory ($\frac{dH}{dt} = 0$), these systems are called conservative, and the trajectory in the state space for a system with n degrees of freedom belongs to a $2n - 1$ manifold defined by $H(\mathbf{q}, \mathbf{p}) = H_0$, where H_0 is the initial energy of the system.

Comparing equations (7) and (5), and considering Cartesian coordinates as the generalised coordinates ($\mathbf{q} = \mathbf{x}$), and the momenta as the velocities ($\mathbf{p} = \mathbf{v}$), we can see that the movement of Braitenberg vehicle 2b corresponds to the movement of a unit mass particle in the potential field $U(\mathbf{x}) = -\frac{1}{2}F(\mathbf{x})^2$. It is worth noting that the initial

conditions of the Braitenberg vehicle are constrained, since the modulus of the initial velocity (v_0) is determined by its initial position $v_0 = F(\mathbf{x}_0)$. Moreover, the trajectory of the vehicle is restricted to lay on regions of zero total energy of the system since $H(\mathbf{x}, \mathbf{v}) = \frac{1}{2}(v_x^2 + v_y^2) + (-\frac{1}{2}F(\mathbf{x})^2)$, and $v = F(\mathbf{x})$. This equivalence has two main consequences for the theory of Braitenberg vehicles. First, because of the equivalence of the two equations, any known theoretical result of Hamiltonian systems is applicable to the movement of vehicle 2b. Secondly, although the interpretation of the movement of a Braitenberg vehicle is rather intuitive, this reformulation enables understanding their behaviour as the movement of a particle in a potential well related to the stimulus the vehicle is immersed in, and, simultaneously, it provides a unified explanation to previous results on the analysis of this vehicles' movement.

Although this equivalence was not found, earlier findings on the closed loop dynamics of Braitenberg vehicle 2b matched known properties of Hamiltonian systems. As pointed out in section 2.1 the motion equations have only saddle points, while a known result from Hamiltonian systems is that they only have stable centres and saddle points, but no attractors. Therefore trajectories on the state space do not converge to an attractor or limit cycle. Interestingly, the above property was proven under more general conditions, i.e. for any d and δ (Raño, 2014b). Another property of Hamiltonian systems is that their behaviour can be chaotic, but this only occurs for systems with at least two degrees of freedom since continuous time chaotic systems require at least a three-dimensional state space. Although in the case of Braitenberg vehicles the state space can be considered four-dimensional, the trajectories are restricted to a three-dimensional manifold of zero total energy. Moreover, the way in which chaos appears in Hamiltonian systems is also well understood (Chirikov, 1960), with local areas of the state space presenting chaotic trajectories as an increasing perturbation is applied to the Hamiltonian function of an initially integrable system. This case was identified empirically for Braitenberg vehicles in (Raño, 2012), where a circularly symmetric stimulus was converted into a parabolic one through the change of a parameter. In the case of the non-chaotic regime, the above work shows that the movement of the vehicle is the combination of non-harmonic periodic oscillations, generating quasi-periodic trajectories that fully cover parts of the workspace, another known result of Hamiltonian systems.

3. Simulation of Braitenberg vehicles

The theoretical background presented proves that Braitenberg vehicle 2b can follow chaotic trajectories, and therefore it can be used for coverage tasks. The question is how to design a stimulus function fulfilling the above assumptions and able to produce chaotic regimes in the behaviour of the vehicle. A sensor based implementation of this vehicle is presented in (Raño, 2014a), where the free area around the robot measured by a laser scanner is used as a stimulus. The work also shows that the trajectories become chaotic if the range of the laser is reduced (i.e. the robot ignores far obstacles) making

the robot ‘bounce’ on the free space boundaries. For a real robot, the free area is easily computed and the solution of equations (1) is the trajectory of the robot in the real world. For a simulated environment, where more experiments can be done, estimating the free area (e.g. through ray-tracing) is computationally expensive. Moreover, we found through simulations that, because the gradient of the free area is required to simulate the motion equations, numerical errors in the computation of the free area strongly affect the accuracy of the numerical integration of equations (1), which for chaotic systems turns out to be a big drawback. Therefore, in our experiments we opted for a stimulus with an analytic expression, for which the gradient can be computed without numerical errors. However, the stimulus is similar to the one used in (Rañó, 2014a) in the sense that the simulated scenarios are topologically equivalent to a real scenario with a free area implementation. The standard way of representing simulated environments is using polygonal boundaries, i.e. ∂D are polygons, yet that would introduce discontinuities on the stimulus gradient if, for instance, we make the stimulus depend on the shortest distance to obstacles. Instead we opted for polynomial boundaries (∂D) since they naturally allow the definition of a smooth stimulus as a function of the position in the environment as follows:

$$S(\mathbf{x}) = \prod_{i=0}^n \beta_i \tanh[\alpha_i P_i(\mathbf{x})] \quad (8)$$

where n is the number of obstacles, and $P_i(\mathbf{x})$ are polynomials such that $\partial D = \bigcup_{i=0}^n \{\mathbf{x} | P_i(\mathbf{x}) = 0\}$. We can choose the polynomials such that $P_i(\mathbf{x}) > 0$ for $\mathbf{x} \in D$ for all i . That generates a positive stimulus in D which will be negative inside the obstacles and outside the outer boundary of the environment. In our case the polynomials are defined through a set of waypoints and their coefficients are obtained by linear regression. The polynomials used in our experiments had degree four and were obtained from eight waypoints of the boundaries of the environment or the obstacles. The hyperbolic tangent was applied to set a bound to the value of the stimulus, just like the free area is bounded by the sensor range. Stimuli defined in this way fulfil the conditions defined in section 2.1, i.e. they are positive in D , vanish in ∂D , but its gradient is not zero for $\mathbf{x} \in \partial D$. The set of parameters α_i control how the stimulus grows close to the boundary in a similar way to how the range of the laser scanner controls how far from the robot an obstacle is detected. For large α values the hyperbolic tangent reaches the flat range faster, just like a short laser range generates a flat free area when there are no obstacles in range. Therefore, if we denote by \bar{l} the range of a laser scanner, there is a mapping from α values to \bar{l} values, i.e. large α values correspond to short laser ranges. To unify the computation of α for all the terms in the stimulus function, and to make it more intuitive, we defined a single distance parameter l from which α is computed for all obstacles, in such a way that the hyperbolic tangent reaches 95% of its final value at distance l from the contour of the polynomial. This allows us to have a single parameter equivalent to the laser range of a mobile robot to run our experiments.

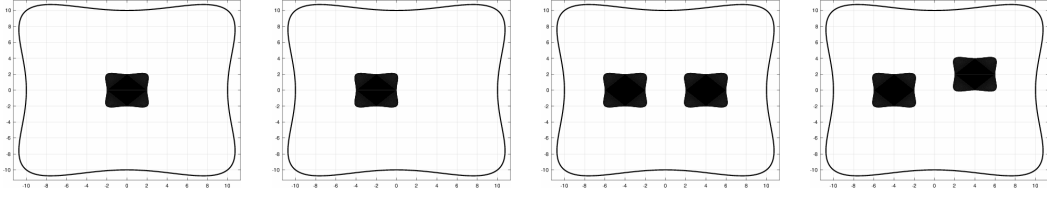


Figure 3. Simple environments used to test symmetry and shape effects on coverage (Scenarios 1 and 2)

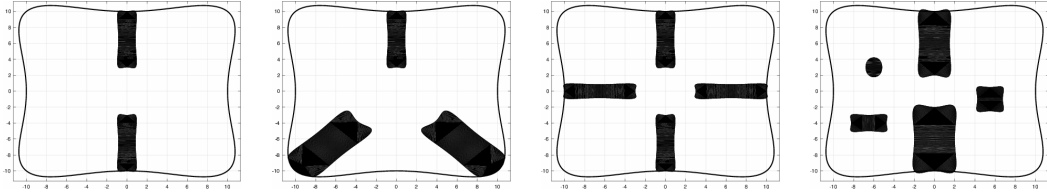


Figure 4. Environments used to test coverage (Scenarios 3)

The scenarios defined above can be simplified to circularly symmetric stimulus, for example defining a circular boundary with (or without) a circular obstacle on its centre. For these stimuli the trajectories have known analytic properties, i.e. they are periodic or quasi-periodic (Rañó, 2012), and chaotic regions appear in the state space as the system's equations are perturbed, e.g. breaking the symmetry by turning the circle into an ellipse or a square, or placing the obstacle away from the centre. Therefore, the (presence/absence of) environmental (i.e. stimulus) symmetry and the shape of the environment (as perturbations of an integrable Hamiltonian system) play an important role in the generation of chaotic trajectories, and, therefore, in the coverage the vehicle will provide. Taking this influence into account we defined several experimental scenarios with different shapes and obstacle configurations to test the coverage provided by this vehicle. However, the shape and symmetry of the stimulus generated by the scenarios do not allow drawing analytic conclusions of the resulting trajectories. The simple scenarios (see figure 3) were selected to test the effect of environment shape and obstacle symmetry on coverage, and they have approximate square and rectangular shapes with different number and location of obstacles. More complex scenarios (shown in figure 4) were designed to test coverage in a more practical way, when the environment corresponds to several connected regions. Although the shape is not exactly square or rectangular, but the solution of a degree four polynomial with way-points in a square or rectangle, we will use the terms square and rectangular to refer to the boundaries of the environment and obstacles. The complete list of scenarios used in this work is labelled as follows:

- Scenarios 0 are obstacle free environments with square (0-s 20×20 distance units) and rectangular (0-r 20×10 distance units) boundaries (not shown in figure 3 because of their simplicity). These are trivial environments to cover but enable evaluating the effect of the boundary shape on coverage and comparing with

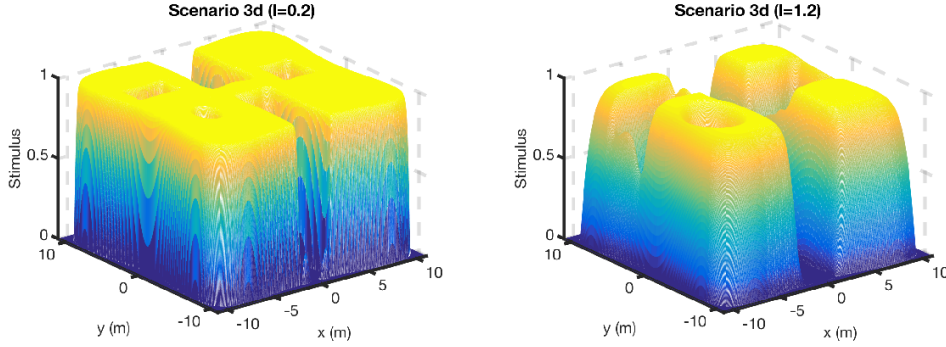


Figure 5. Stimulus generated by scenario 3d for $l = 0.2$ (left) and $l = 1.2$ (right).

environments with similar symmetry properties (scenarios 1a).

- Scenarios 1 have the same boundary as the previous ones, but contain one square (1x-s) or rectangular (1x-r) obstacle, which, as the two leftmost plots of figure 3 shows, can be centred (1a-x) in the environment generating a symmetric stimulus (but not circularly symmetric, i.e. no analytic results apply), or not centred (1b-x). Therefore, results from scenarios 0 are expected to be similar to the symmetric environments. There are four scenarios 1 combining the obstacle at the centre of the environment vs. not centred, and square vs. rectangular boundaries/obstacles.
- Scenarios 2 contain two obstacles with different configurations as shown in the two rightmost plots of figure 3. These scenarios also come in the square (2x-s) and rectangular (2x-r) variants, with some symmetry (2a-x) and asymmetry (2b-x) in the obstacle location. Although for these environments the stimulus radial symmetry is lost (stimuli can have several local maxima), they constitute a stepping stone towards the more complex environments (scenarios 3).
- Scenarios 3 consist of a square environment with rectangular obstacles separating different simply connected areas (or ‘rooms’) as shown in figure 4. The gap between these areas allows the vehicle to cross from one to another with high probability for some range of small l values (when the stimulus value remains high close to the obstacles). We will denote these by scenario 3a, 3b, 3c, and 3d (from left to right in figure 4), where, as it can be seen, scenario 3d contains three obstacles with different shapes in the two simply connected areas. Figure 5 shows the stimulus function for scenario 3d for two different sensor range parameters ($l = 0.2$ and $l = 1.2$). For a small l value, the stimulus function between the two main areas of the environment is flat and has a similar value to the value in the two regions. However, for a longer sensor range ($l = 1.2$) the stimulus shows several valleys close to pairs of obstacles (like the passage between the two ‘rooms’), which reduces the likelihood of the vehicle moving to these areas.

In all the scenarios simulated the connecting function for the Braitenberg vehicle was simply a linear function with positive slope, $F(s) = v_M s$, with $v_M > 0$ being the

maximum speed selected to be 0.5 m/s at the stimulus maximum value. It is worth noting the following two effects the change of the parameter l has in the trajectories of the vehicle. First, because the forward speed is $v = F(S(\mathbf{x}))$, and we selected $F(s)$ to be linear, the plots presented in figure 5 represent scaled versions of the forward velocity of the vehicle at each point of the environment. Clearly, increasing l reduces the average speed of the vehicle – fewer points in the right stimulus plot have values close to 1 –, which for the same simulated time means stimulus with larger l will generate a shorter trajectory. Secondly, because Braitenberg vehicle 2b moves towards areas of high stimulus value, the probability of the vehicle moving between the two simply connected regions (rooms) is smaller for large l values as the stimulus on the passage has a smaller value. To evaluate the coverage in each scenario for different values of the parameter l , we performed two sets of experiments presented in the rest of this section. First, to ensure the trajectory densely covers the workspace we used a common measure found in the dynamical systems literature, the dimension of the trajectory, specifically the box counting dimension. A trajectory generated by a dynamical system would typically be one-dimensional, but for chaotic systems this dimension can be fractional and higher than one. Moreover, if a trajectory is dense in some subspace of the state space, it should have the dimension of the subspace, in the case at hand the dimension should be two for the vehicle to cover the workspace. Therefore, a condition for coverage is that the dimension of the trajectory generated by the dynamical system (1) will be as close as possible to two. This is a necessary condition, but it does not ensure that the vehicle reaches the whole workspace, since the trajectory could be dense only in a small region of the area to be covered. Therefore, a sufficient condition, which we tested in the second set of experiments, consist of measuring the actual area the vehicle visits for different l values. This is a standard measure in the coverage literature, but it presents the limitation that the environment has to be discretised (typically with the size of the robot), and therefore it is simply an approximation of the coverage. Despite this drawback we will use this measure as the sufficient coverage condition in this work.

3.1. Dimensional analysis of the trajectories

Dimensional analysis of dynamical systems' trajectories is typically performed on scalar time sequences of experimental data to estimate the dimension of chaotic attractors. The, so called, fractal dimension, measures the complexity of sets – e.g. the solution of the dynamical system time series –, but also how this set fills the space in which it lives. While the fractal dimension of standard geometric shapes takes integer values, a chaotic attractor can have a non-integer fractal dimension, typically smaller than the dimension of the underlying state space. In our case, if the dimension of the trajectories is close to two, the vehicle will wander on a surface embedded in its three-dimensional state space. As we saw, this can be the case of Hamiltonian systems, since trajectories are restricted to the manifold of constant total energy. Several methods can be used to calculate fractal dimensions (Sprott, 2003), yet probably the most common one is the

capacity (or box-counting) dimension (Russell, Hanson, & Ott, 1980). To compute the box-counting dimension one can start by considering a parallelotope (a d -dimensional cube) enclosing the set for which the dimension has to be computed. This parallelotope is split into smaller cubes of side length s , therefore, for a fixed length parallelotope, the total number of small cubes is proportional to $1/s^d$. For a given set A , the number of small cubes (boxes) containing points of A is a function of the box size $N(s) \propto 1/s^d$. The capacity dimension is defined as the limit:

$$d = \lim_{s \rightarrow 0} \frac{\log N(s)}{\log(1/s)} \quad (9)$$

where $N(s)$ is the number of hyper-cubes containing points of the set A . Instead of numerically computing this limit it is better to estimate the slope of $\log N(s)$ as a function of $\log(1/s)$ for several orders of magnitude of $\log(1/s)$.

Since we are interested in area coverage, whilst our dynamic systems is three-dimensional, we will first project the state space trajectories to the two-dimensional workspace and consider only the x and y coordinates to measure coverage. This means that it is not relevant the heading with which the vehicle reaches a point of the environment, and simplifies the computational requirements of the dimension estimation algorithm. We used a 2D grid made of square tiles of side s which contain the $x - y$ projected trajectory, then the algorithm counts the number of boxes containing at least one trajectory point and compares that to the number of obstacle-free tiles that make up the grid. Because the box counting dimension is computed over a trajectory which depends on the initial conditions we performed 25 simulations with random initial conditions in the free space for each scenario listed in section 3, and each l value tested. To analyse the effect of the parameter l over the trajectory dimension, we ran the 25 simulations for sixteen different values of l from 0.01 to 8.5, with each simulation lasting 30000 seconds, and the trajectory was sampled 75 times per second, producing over two million points as the set to compute the box counting dimension. This high precision was selected to ensure consecutive trajectory points are close enough so that the variation range of the parameter s spawned several orders of magnitude. The dimension was then computed as the slope of the linear regression using the values of $\log(1/s)$ and $\log N(s)$ in the 55% middle range of $\log(1/s)$ to avoid border value effects of s . From the 25 simulations obtained for each value of l (with random initial conditions) we computed the average, the minimum, and the maximum dimension and plotted it against the parameter encoding the sensor range. These plots are presented in figure 6 for the defined scenarios (except for scenarios 3 where only the maximum and average dimensions are shown to make the plot less cluttered), where the solid lines correspond to the maximum, the dashed lines to the average, and the dotted lines to the minimum dimensions.

Before proceeding with the analysis of the dimensional results it is worth noting that the estimate of the dimension depends on several factors like the length and sampling rate of the trajectory, and the way of computing the slope from the discretised boxes.

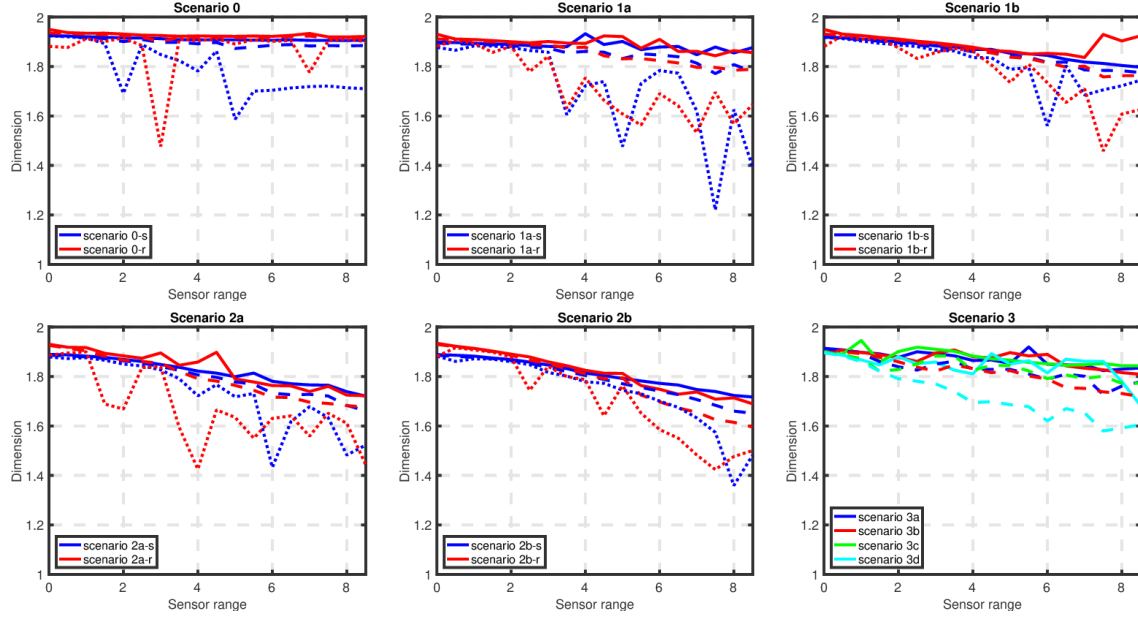


Figure 6. Trajectory dimensions as a function of the sensor range for all experimental scenarios.

Therefore, the fact that some trajectories have dimensions lower than 2 can be a result of the simulation time being too short for that specific set of initial conditions, or the selection of points in the computation of the slope. Attending to the average dimensions shown on the plots (figure 6), we can see that most of the trajectories have a dimension above 1.6 (see average dimensions, dashed line in the figure) regardless of the value of l , and the dimension stays above 1.8 for $l < 4$. This clearly shows that the coverage provided by the vehicles is dense. Moreover, for all the environments simulated, the trajectories generated by the Braitenberg vehicle for small values of l (approximately $l < 2$) have a dimension close to 2, indicating excellent coverage (although it might be simply local coverage as the dimension is only a necessary condition). If we consider the differences between the maximum, minimum and average dimensions in the plots, we can also conclude that low dimension trajectories are the exception, since the average dimension graphs (dashed lines) are typically close to the maximum dimension graphs (solid lines). Because the trajectories are obtained from random initial positions in the environments, the distribution of the dimension value for the trajectories of a given scenario are skewed towards high dimensional values. Similarly, the probability of an initial point in the environment generating low dimension trajectories is small.

As already stated, the shape of the environment can be considered as a perturbation from the circularly symmetric case, for which a closed-form solution of the differential equations (1) exist. Regions of chaotic regime can then appear changing the dimension of the resulting trajectories. In this case, however, the analysis of the trajectory dimension for different shapes has no noticeable effect (see maximum and average dimensions for each individual plots in figure 6), as none of the shapes – square and rectangular –

have closed form solution for the motion equations. On the other hand, the effect of asymmetry seems more noticeable if we compare the plots corresponding to scenarios ‘a’ and ‘b’, as the dimension seems to decrease more for large l values for non-symmetric scenarios. This trend can be clearly seen by comparing the maximum dimensions for scenario 1a (one centred obstacle) and 1b (non-centred obstacle), and to a less extent for the two obstacle case (scenarios 2a and 2b). Interestingly, comparing the plots of the simple scenarios we can conclude that the number of obstacles has a higher impact in the dimension than any of the other factors, something that could not be predicted theoretically. The average dimension for large values of l falls as the number of obstacles increases, as can be seen from the comparison of the plot for scenarios 0 (no obstacle), 1 (one obstacle), and 2 (two obstacles). A similar effect appears in the complex scenarios defined. While scenarios 3a, 3b, and 3c display a similar average dimension for the trajectories, scenario 3d (which includes three obstacles) has a significantly lower dimension as the value of the parameter l , the sensor range, becomes larger. Another interesting point to note from these results is the similarity in the dimension analysis between simple and complex scenarios (3a-3c). In fact the first three complex scenarios defined are topologically equivalent to scenario 0, since the number of obstacles (holes in the stimulus function) is zero.

3.2. Area coverage analysis

The analysis of the dimension of the trajectories is a good measure of how well the vehicle covers a given area, moreover, it can be applied to any type of robot trajectory, from randomly generated to planned (given a discretised workspace). However, even if the trajectory dimension is close to two, the area covered by the vehicle does not necessarily correspond to the whole workspace. The way used in the literature to measure coverage consists on, given the size of the robot, measuring the actual area covered in a discretised version of the workspace. Normalising this measure by the total free area provides an even better measure than the area itself, although due to the discretisation process, this is just an estimate, since some discretised tiles could have mixed regions of free and occupied space. We considered a discretised cell to be free if the centre was free, so the total area to cover was also an approximate. We also followed this approach to evaluate coverage assuming the vehicle was 0.05 length units (standard environment sizes were 20 length units), and we stored the evolution of the (normalised) covered area over time. This normalised area provides an estimate of the necessary coverage condition, and to explore the effect of the sensor range on the coverage we ran 25 new simulations lasting 500 minutes each for each one of the scenarios and each sensor range parameter used in the previous section. The starting pose of the vehicle was selected randomly in the free space, so the conditions of the two experiments were alike.

Figure 7 shows the maximum covered area for the scenarios 0, 1, and 2 as a function of the range parameter and the simulation time. Although dense trajectories cover the space in the limit of time going to infinity, from a practical point of view (given

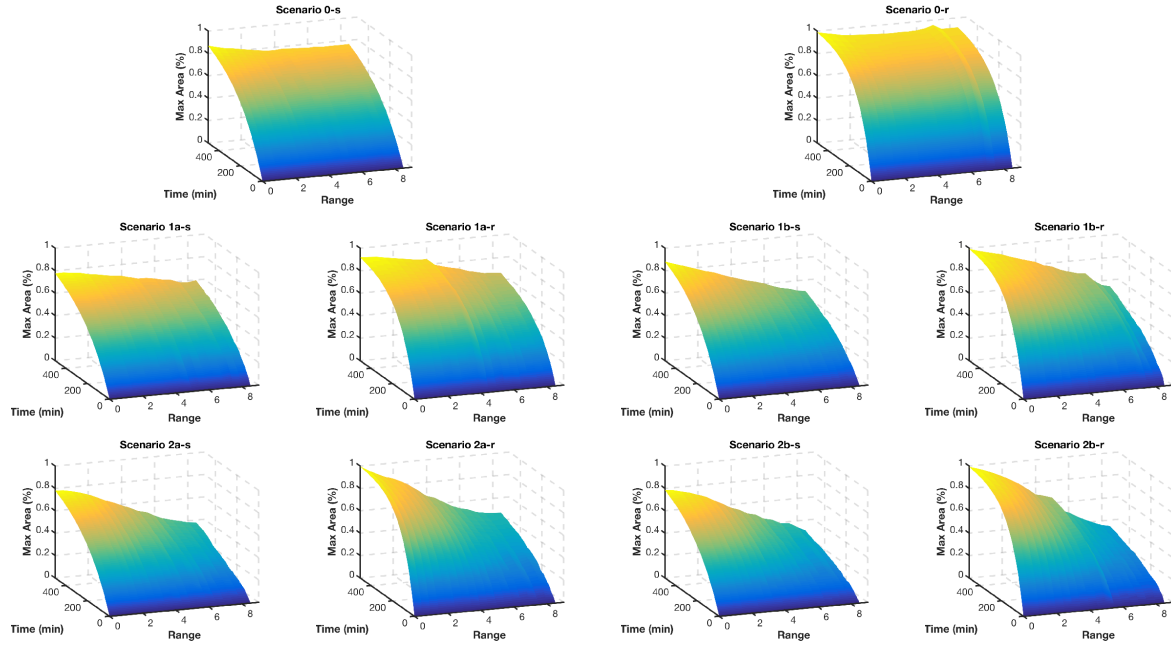


Figure 7. Maximum covered area as a function of the sensor range and simulation time for the simple scenarios (scenarios 0, scenarios 1, and scenarios 2)

the environment discretisation) full coverage could be achieved in finite time for some environments and some parameters of the Braitenberg vehicle. It is worth noting that the coverage speed is a function of different factors like; the definition of the stimulus (the environment and sensors used), the ratio between the wheelbase and the distance between sensors (set to 1 in this work to obtain a Hamiltonian system), the function connecting the sensors and the motors $F(s)$ (the controller), and the initial pose of the vehicle. However, here we are mainly interested in the stimulus (symmetry and shape), and in showing that coverage is also possible when the dynamical system (1) has no closed-form solution. An interesting result from the simulations in these scenarios is the correlation between the trajectories' average dimension and the maximum coverage as a function of l . Although for small values of l the trajectories of the vehicle cover between 80% and 100% of the area (see plots in figure 7), as l increases the area covered drops for the same scenarios the average dimension drops (all but scenarios 0, cf. figure 6).

Another interesting experimental result comes from analysing the effect of the different shapes of environments on the coverage, as scenarios with rectangular boundaries always reach a higher coverage ratio, more noticeable for small l values, compared to their square counterparts. In fact, most of the rectangularly bounded environments reach close to full coverage in the simulated time for the lower sensor range simulated, $l = 0.01$. Moreover, the slope of the graphs over time, the derivative of the normalised area covered w.r.t. time, is higher than the square bounded environments (e.g. cf. top row plots on figure 7), which means that the coverage is also faster for rectangular environments. This confirms that there is indeed an effect of the shape

over the coverage as stated earlier, and not only for the case of trajectories with closed-form vs. not closed-form solution. The symmetric or asymmetric location of obstacles appears to have no (or minimal) influence on the covered area. Whilst in the environments containing one obstacle the coverage seems to improve slightly for the non-centred obstacle (cf. scenario 1a-s vs. ab-s, and 1a-r vs. 1b-r), in the scenarios with two obstacles there is no apparent effect. Finally, as we observed in the dimensional analysis, the number of obstacles also has an effect on the covered area, specially for large values of l . By looking at the largest simulated values of l , we can see that the area covered for each environment in the simulated time decreases as the number of obstacles increases (including the no obstacle case, scenarios 0). This can be qualitatively explained by the fact that Braitenberg vehicles perform a (non-holonomic) hill climbing on the stimulus. Indeed, unless the stimulus between two obstacles has a distinct peak, the vehicle will head towards other areas of the environment with higher stimulus values (i.e. it will try to reach the point with larger clearance).

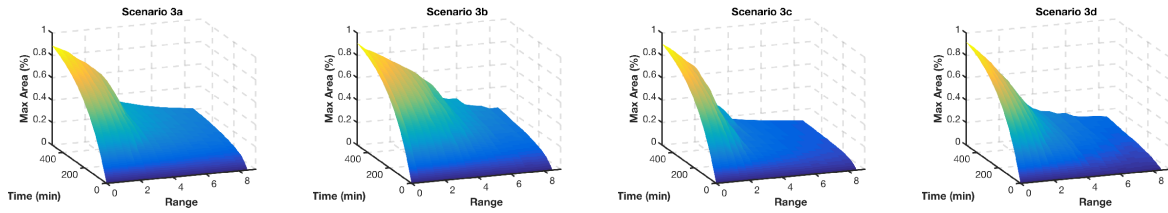


Figure 8. Maximum covered area as a function of the sensor range and time for scenarios 3

Figure 8 shows the maximum areas for scenarios 3 (see figure 4). The plots show that the maximum covered area after the simulated time is around 90% for short sensor range parameter, and continuously falls as l increases. Because the environments consist of connected regions separated by narrow passages, one would expect the vehicle to be able to cross between the ‘rooms’ for small values of l , while – for larger values of l – the vehicle can perceive the sides of the passages – generating a lower free area around, i.e. stimulus value – and making it turn and stay in the same ‘room’. This reduction of the stimulus on the passage can be clearly seen in figure 5. For large sensor range the vehicle covers only the ‘room’ in which the initial condition of the simulation was. The range sensor distance l for which the covered area falls below the area of the individual ‘rooms’ depends on the size of the opening. For instance, in scenario 3b the opening is larger than in scenario 3c, which is reflected in the corresponding plots of figure 8. At the large end of l values, we can observe that the coverage is quite poor, reaching only around 20% of the total area. However, even in more complex environments like scenario 3d, for small values of the sensor range coverage is still around 90%, which indicates the vehicle can cover arbitrarily complex environments.

4. Comparison with Random Motion for Coverage

This section presents experiments performed with different types of random motion for coverage as a comparative way to evaluate chaotic coverage of the Braitenberg vehicle controller. The trajectories, in these cases, consist of sequences of straight line segments (inside the free area) with random length, and bounded by the distance to the nearest obstacle in front of the robot. Specifically, we tested trajectories generated using two distributions for the forward distance travelled by the vehicle (the segment length), the uniform and Levy distributions. Because the Levy distribution is heavy tailed, i.e. there is no limit to the maximum distance generated, we kept generating random segment lengths until the end point was inside the free space, which means the actual distribution is truncated. The motion directions were also chosen randomly following a uniform distribution in the range $(-\pi, \pi]$, and the robot turned at the endpoint of each segment. There are two important issues to note. First, because the coverage for a fixed time interval strongly depends on the forward velocity of the vehicle (the faster it moves the more area it can cover), we set the velocity to be the same as the velocity a Braitenberg vehicle would follow at each point, i.e. the instantaneous forward velocity is the same function of the position in the environment $v = F(S(\mathbf{v}))$. This allows a fair comparison between the coverage provided by the Braitenberg vehicle and the random motion controllers. Secondly, because the Braitenberg vehicle follows a unicycle model, i.e. non-holonomic motion, while random trajectories are typically implemented following an integrator model, i.e. holonomic motion, in the experiments a non-holonomic vehicle was simulated, and the random on-the-spot turns were performed with a turning speed dependent on the forward velocity. The analysis of early experiments immediately showed a very low coverage performance in terms of both dimension and area for random movement. This effect was already pointed out in (Nakamura & Sekiguchi, 2001) where chaotic motion was shown to outperform Brownian motion, the reason being the time “wasted” by the robot to turn on the spot, which for chaotic coverage is used to turn while moving. Hence, we decided to perform experiments where the vehicle was instead holonomic, i.e. it did not need to turn but could move in a new direction upon completion of a straight line movement.

The experimental procedure for these experiments is the one followed for the simulation of Braitenberg vehicles (section 3), i.e. 25 trajectories of 500 min were simulated for different values of l with random starting positions in each scenario. The box-counting dimension was computed from the trajectories using the same technique as before. Figure 9 shows, for each scenario, the plots of the maximum dimension among the 25 simulations for the different random trajectory generation mechanisms. As a quick comparison between figures 9 and 6 shows, the dimensions of the trajectories generated by the Braitenberg vehicle controller are consistently higher throughout all the scenarios. Moreover, although it is not represented in figure 9, the variability in the dimension for the non-holonomic random walks is significantly higher, which makes the maximum dimension in this case not very representative of the trend of the real coverage

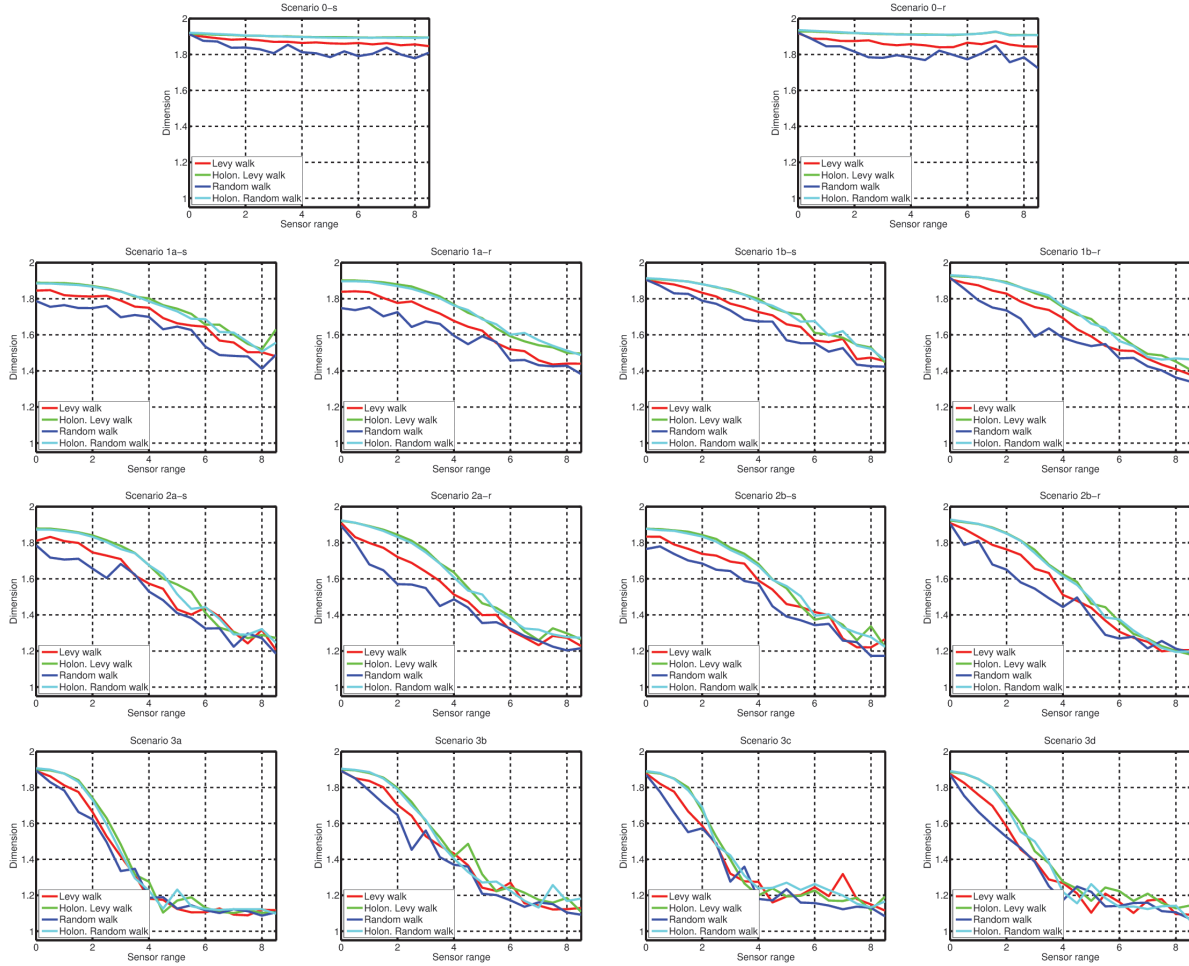


Figure 9. Trajectory maximum dimensions as a function of the sensor range for random movements.

provided (but enables a “best-case” comparison). Because these trajectories are not generated by a Hamiltonian system, we do not expect the shape of the environment nor the location of the obstacles having an effect over the trajectories. However, the plots in figure 9 show a clear decrease in the dimension as the number of obstacles increases (cf. plots on the three first rows of the figure). This effect also appeared in the dimension of the Braitenberg vehicle controller, yet for the random motion controllers it is much more pronounced. Considering only the random motion controllers, it is worth noting that the dimension of the Levy walk is consistently higher than the dimension of the random walk controller (i.e. uniform distribution) for the non-holonomic vehicle. This is not the case for the holonomic vehicles, as both plots overlap for many of the scenarios, although it could be the effect of truncating the probability for the Levy walk. Finally, the complex scenarios (3a-3d) show a quite a similar profile for the dimension regardless of the number of ‘rooms’ and whether there are obstacles in them. Moreover, for large values of the sensor range the maximum dimension of all the trajectories is around 1.2, much lower than 1.8 for the trajectories followed by the Braitenberg vehicle. In

summary, attending to the dimension of the trajectory generated, the best performance (higher dimension) of a random controller is provided by holonomic vehicles, but they are outperformed by a chaotic Braitenberg controller.

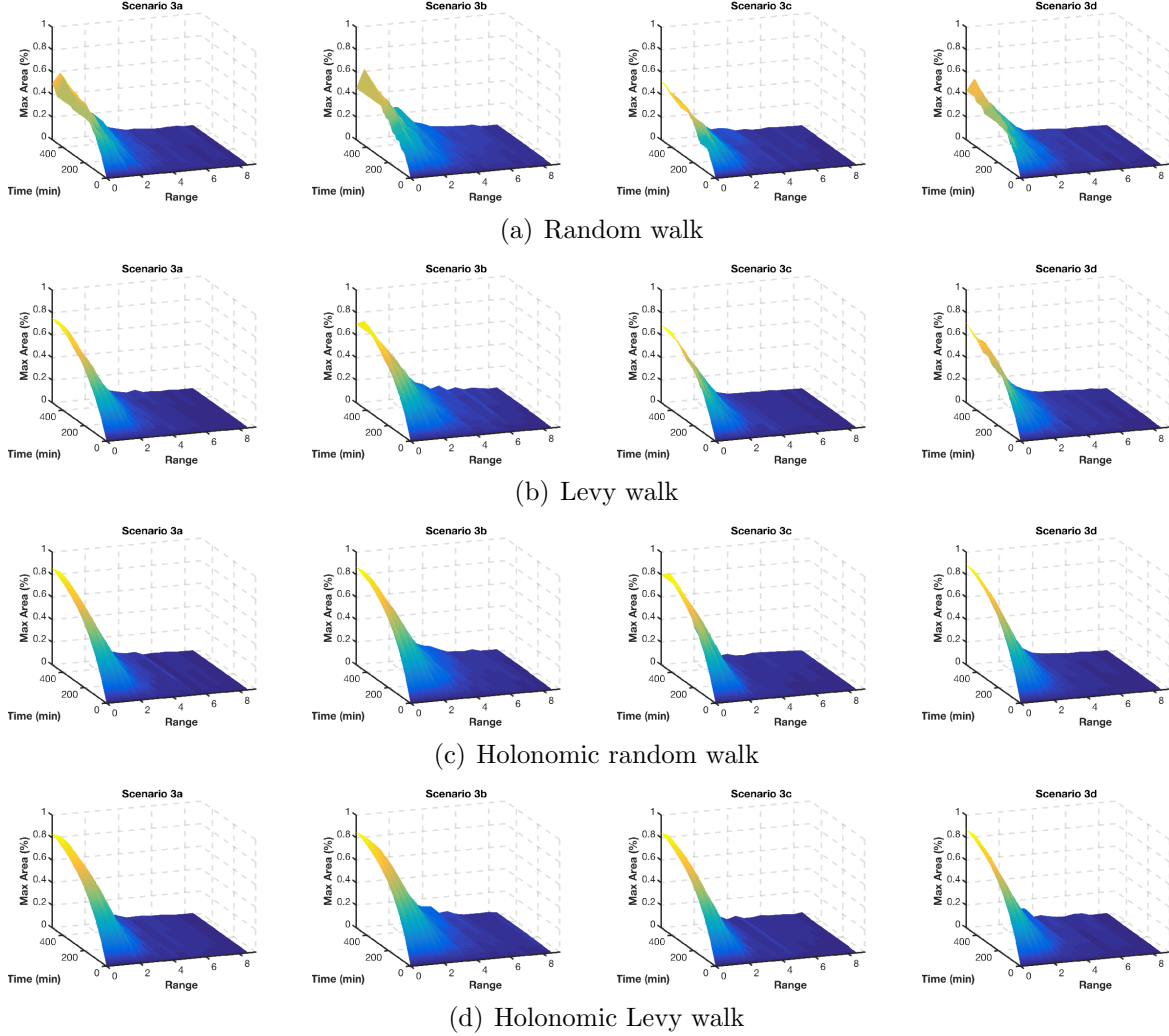


Figure 10. Maximum covered area as a function of the sensor range and time for scenarios 3

Like in the case of the Braitenberg vehicle controller, the trajectory dimension is just a necessary condition which measures how dense the coverage is. Therefore, we also measured the evolution over time of the covered area for the random motion controllers as a function of l , which in this case affects only the forward speed of the robot, and the turning speed for the non-holonomic vehicles. Obviously, a result to be expected is that holonomic robots will cover more area for any scenario given the fixed simulation time and the velocity profile. Since these robots do not need to turn in order to change their direction, they do not spend additional time at the end of each segment, but instead use that time to carry on with the coverage. This effect can be observed in figure 10, if we compare figure 10(a) with 10(c), and figure 10(b) with 10(d), which shows the

results of the area covered for the complex environments. Experimental results showed no significant difference between the Levy and uniformly distributed random controllers for holonomic vehicles (cf. figure 10(c) and 10(d)). However, for non-holonomic vehicles Levy walks performed slightly better than random walks in terms of the covered area, specially for small values of the sensor range. Although figure 10 shows only the results for scenario 3, these conclusions apply to all other scenarios. Comparing these results with the corresponding area evolution of the Braitenberg controller, figure 8, we see that the coverage provided by random motion controllers is always below the proposed chaotic controller. Only randomly moving non-holonomic robots show a similar performance for low values of l .

5. Conclusions and Further Work

This paper presents an alternative approach to area coverage in robotics based on the chaotic movement of a biologically inspired mobile robot controller. Through a mixture of theoretical and experimental results, our solution to the coverage problem exploits chaos to generate dense trajectories in the workspace to cover. This technique falls between heuristic and complete methods, as in theory chaotic trajectories are dense, but potentially only in a region of the whole workspace. We presented a theoretical proof that a vehicle controlled using the Braitenberg vehicle 2b is driven by a Lorenz force, and, under some assumptions, it is a Hamiltonian system equivalent to a friction-less mass in a potential well. This result supports the use of such a controller to generate dense coverage trajectories. For a carefully designed stimulus, energy arguments can prove collision-free trajectories in the environment. Simulated experimental results show that trajectories resulting from the closed loop system are indeed dense, i.e. they have a high fractal dimension, and cover the whole workspace when the parameter corresponding to the perceptual range of the vehicle is small. A similar result was found in the existing heuristic coverage literature (Doty & Harrison, 1993) and is exploited in commercial randomly moving vacuum cleaners, which typically have short range sensors. We also proposed using the dimension of the vehicle trajectory, the box-counting dimension (borrowed from the literature of the analysis of chaotic dynamical systems), as a necessary measure for coverage. A comparison with several random movement controllers showed that to achieve a performance similar to our chaotic controller a holonomic motion system is required. From a practical viewpoint, the complexity of implementing both controllers, Braitenberg vehicle or random motion, is similar, yet a holonomic motion system is mechanically more complex. Non-holonomic robots with a random motion controller perform much worse than the Braitenberg vehicle. This result can be understood from a theoretical perspective, since the dynamical system describing the motion has equilibrium points everywhere in the environment boundary containing unstable manifolds. This means that the turning velocity of the vehicle moving away from an obstacle increases exponentially – as a first order approximate – while it still moves forward (i.e. it never stops).

Despite experimentally showing that coverage is achieved using this controller, as for other heuristic methods, the robot will very likely cover some areas several times before reaching others. This means the coverage is not optimal in terms of time or energy spent. In this work we assumed the sensor range to be fixed over trials, but more efficient coverage could result from dynamically adjusting this range over time as a function of the environment trying to avoid multiple passes over the same area. Hence, future work could focus on the design of an automatic range adjustment technique, which would adapt the sensor range according to the minimum distance between obstacles in the environment. This automatic adjustment would be highly beneficial for applications like surface cleaning, where optimality conditions can be clearly defined in terms of time or energy spent. Yet our bio-inspired controller has very interesting properties for indoor surveillance tasks, as the chaotic nature of the movement can make trajectories difficult to predict. Because of its computational simplicity, this technique could also be used for multi-robot (swarm) coverage, where each robot is considered by the rest as a moving obstacle. This would increase the dimension of the dynamical equations describing the system behaviour, and would include coupling between robots, but it would certainly reduce the time necessary for complete coverage. Alternatively, a stochastic approach (Rañó, Khamassi, & Wong-Lin, 2017) to Braitenberg vehicles could be used to characterise swarm coverage, which, in turn, could help solving a limitation of this work, namely finding a formal proof of complete coverage.

References

- Acar, E., Choset, H., & Lee, J. (2006). Sensor-based coverage with extended range detectors. *IEEE Transactions on Robotics*, 22(1), 189-198.
- Acar, E., Choset, H., Rizzi, A., Atkar, P., & Hull, D. (2002). Morse decompositions for coverage tasks. *The International Journal of Robotics Research*, 21(4), 331-344.
- Bircher, A., Kamel, M., Alexis, K., Burri, M., Oettershagen, P., Omari, S., ... Siegwart, R. (2016). Three-dimensional coverage path planning via viewpoints resampling and tour optimization for aerial robots. *Autonomous Robots*, 40, 1059-1078.
- Braitenberg, V. (1984). *Vehicles. experiments in synthetic psychology*. The MIT Press.
- Chirikov, B. (1960). Resonance processes in magnetic traps. *Journal of Nuclear Energy Part C: Plasma Physics*, 1(4), 253.
- Choset, H. (2001). Coverage for robotics – a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31, 113-126.
- Doty, K., & Harrison, R. (1993). Sweep strategies for a sensory-driven, behavior-based vacuum cleaning agent. In *Proceedings of the aaai 1993 fall symposium series instantiating real-world agents research triangle park* (p. 1-6).
- Fraenkel, G., & Gunn, D. (1961). *The orientation of animals. kineses, taxes and compass reactions*. Dover publications.
- Giszter, S., Mussa-Ivaldi, F., & Bizzi, E. (1993). Convergent force fields organized in the frog's spinal cord. *The journal of neuroscience*, 13(2), 467-491.

- Hameed, I., Cour-Harbo, A. I., & Osen, O. (2016). Side-to-side 3d coverage path planning approach for agricultural robots to minimize skip/overlap areas between swaths. *Robotics and Autonomous Systems*, 76, 36-45.
- Harper, N., & McAlpine, D. (2004). Optimal neural population coding of an auditory spatial clue. *Nature*, 430, 682-686.
- Hooper, S. (2001). Central pattern generators. In *Encyclopedia of life sciences*. John Wiley & Sons.
- Ijspeert, J. (2008). Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4), 642-653.
- Ijspeert, J., Nakanishi, J., Hoffmann, H., Pastor, P., & Schaal, S. (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation*, 25(2), 328-373.
- Latombe, J. (1991). *Robot motion planning*. Kluwer Academic Publishers.
- Lebastard, V., Boyer, F., Chevallereau, C., & Servagent, N. (2012). Underwater electro-navigational in the dark. In *Proceedings of the international conference on robotics and automation (icra)* (p. 1155-1160).
- Li, C., Song, Y., Wang, F., Wang, Z., & Li, Y. (2016). A bounded strategy of the mobile robot coverage path planning based on lorenz chaotic system. *International Journal of Advanced Robotic Systems*, 13(3).
- Lilienthal, A. J., & Duckett, T. (2004). Experimental analysis of gas-sensitive Braitenberg vehicles. *Advanced Robotics*, 18(8), 817-834.
- Lovász, L. (1993). Random walks on graphs. In *Combinatorics. paul erdős is eighty* (Vol. 2, p. 1-46).
- MacKenzie, D., & Balch, T. (1996). Making a clean sweep: Behaviour based vacuuming. In *Aaai fall symposium, instationating real-world agents*.
- Nakamura, Y., & Sekiguchi, A. (2001). The chaotic mobile robot. *IEEE Transactions on Robotics and Automation*, 17(6), 898-904.
- Nori, F., & Frezza, R. (2005). A control theory approach to the analysis and synthesis of the experimentally observed motion primitives. *Biological Cybernetics*, 93(5), 323-342.
- Ohashi, S., Morimoto, T., Suzuki, Y., Miyakawa, H., & Aonishi, T. (2014). A novel behavioral strategy, continuous biased running, during chemotaxis in *drosophila* larvae. *Neuroscience Letters*, 570, 10-15.
- Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*(8), 651-668.
- Paraschos, A., Daniel, C., Peters, J., & Neumann, G. (2013). *Probabilistic movement primitives*.
- Porter, J., Craven, B., Khan, R., Chang, S., Kang, I., Judkewitz, B., ... Sobel, N. (2007). Mechanisms of scent-tracking in humans. *Nature Neuroscience*, 10(1), 27-29.
- Qian, N. (1997). Binocular disparity and the perception of depth. *Neuron*, 18, 359-368.

- Rajan, R., Clement, J., & Bhalla, U. (2006). Rats smell in stereo. *Science*, *311*, 666-670.
- Raño, I. (2012). A systematic analysis of the braitenberg vehicle 2b for point like stimulus sources. *Bioinspiration & Biomimetics*, *7*(3).
- Raño, I. (2014a). The bio-inspired chaotic robot. In *Proceedings of the ieee international conference on robotics and automation*.
- Raño, I. (2014b). Biologically inspired navigations primitives. *Robotics and Autonomous Systems*, *62*(10), 1361-1370.
- Raño, I., Khamassi, M., & Wong-Lin, K. (2017). A drift-diffusion model of biological source seeking for mobile robots. In *Proceedings of the ieee international conference on robotics and automation*.
- Russell, D., Hanson, J., & Ott, E. (1980). Dimension of strange attractors. *Physical Review Letters*, *45*, 1175-1178.
- Salumäe, T., Raño, I., Akanyeti, O., & Kruusmaa, M. (2012). Against the flow: A braitenberg controller for a fish robot. In *Proceedings of the international conference on robotics and automation (icra)* (p. 4210-4215).
- Schaal, S. (2006). Dynamic movement primitives -A framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines* (p. 261-280). Springer.
- Schroeder, A., Ramakrishnan, S., Kumar, M., & Trease, B. (2017). Efficient spatial coverage by a robot swarm based on an ant foraging model and the lévy distribution. *Swarm Intelligence*, *11*, 39-69.
- Shnaps, I., & Rimon, E. (2016). Online coverage of planar environments by a battery powered autonomous mobile robot. *IEEE Transactions on Automation Science and Engineering*, *13*(2).
- Sprott, J. (2003). *Chaos and time-series analysis*. Oxford University Press.
- Volos, C., Kyprianidis, I., & Stouboulos, I. (2012). A chaotic path planning generator for autonomous mobile robots. *Robotics and Autonomous Systems*, *60*, 651-656.
- Webb, B. (2001). A spiking neuron controller for robot phonotaxis. In *Biorobotics* (p. 3-20). The MIT/AAAI Press.
- Wong, S., & MacDonald, B. (2003). A topological coverage algorithm for mobile robots. In *Proceedings of the 2003 ieee/rsj international conference on intelligent robots and systems* (Vol. 2, p. 1685-1690).
- Yang, X., Patel, R. V., & Moallem, M. (2006). A Fuzzy-Braitenberg Navigation Strategy for Differential Drive Mobile Robots. *Journal of Intelligent Robotic Systems*, *47*, 101-124.