Contemporary Engineering Sciences, Vol. 10, 2017, no. 25, 1223 - 1232
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ces.2017.79110

A Study on Machine Learning Models for Convergence Time Predictions in Reactive Navigation Strategies

Fredy Martínez, César Hernández and Angélica Rendón

District University Francisco José de Caldas
Technological Faculty, Bogotá D.C., Colombia

Copyright © 2017 Fredy Martínez, César Hernández and Angélica Rendón. This article is distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

A set of finite automatons (robots) have been used by the ARMOS research group in a series of navigation experiments in controlled environments under the principle of autonomous reactive control. The principle of navigation is supported in the ability of the dynamic billiards to sweep an entire environment in a finite time. The dynamic properties of the billiards have been replicated in the robots, adding to the navigation strategy behavioral policies that control the development of tasks. Robots have no a priori knowledge of the environment topology or its location within it. In addition, given the presence of other robots, the environment is considered dynamic. Previous research has demonstrated the navigability of the strategy, as well as its ease of implementation in robots with low processing capacity. This work focuses on the development of models that allow predicting the navigation time of robots under pre-established conditions. Observing the dependence of time with the topology of the environment, the capacity of the robot and the number of robots, this research proposes a model based on a Long Short-Term Memory (LSTM) network. The results against data not used in the training validate the performance of the model.

Keywords: autonomous robot, convergence time, deep learning, path planning

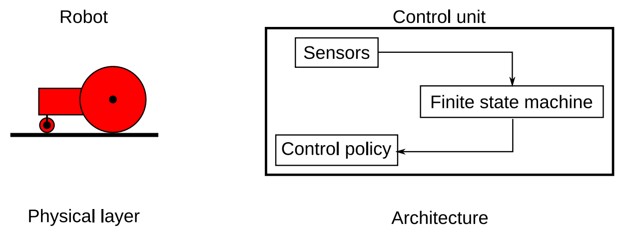

Figure 1: Abstraction architecture for planning and control

1 Introduction

A finite automaton (FA) or finite state machine is a computational model that automatically computes an input to produce an output. The possible response outputs are defined within a finite set of states from which the control selects one according to the received stimulus or input. This simple processing structure has been widely used to implement autonomous robots of low complexity, in particular in exploration and navigation tasks in unknown environments [7]. These tasks can be developed assuming a geometric configuration of the environment on which a route is analyzed and traced (central control unit), or giving to the robot the possibility of reacting specifically to local readings in the environment (control over the robot) [9, 8]. From this basic principle of movement, it is possible to program the robots to the development of tasks much more complex, maintaining the simplicity of design [10].

The ARMOS research group has conducted multiple autonomous navigation experiments with very simple robots inspired by dynamic billiards and wild motion (Fig. 1) [5]. The basic principle of movement of these robots is supported in the ability of the dynamic billiards to sweep an environment in a finite time. During the experiments the navigation environment is divided into regions with the intention of coordinating the movement of the robot from region to region. This control of movement is made by means of events recognizable by the robot [2]. If the robot recognizes a particular event, then it changes its behavior and enters a new region. This type of design allows to discretize the system and model it as a hybrid system, in which events trigger discrete actions, but the movement in each region is continuous [1, 4].

A very interesting case is the control of a swarm of robots [8]. The principle of operation of these swarms is based on the emergent behavior of the system because of the interaction of the robots with the environment [6]. The swarm of robots becomes a very attractive alternative when the designer wants to increase the robustness of the robotic system [13]. This is because the robots continue to develop their task even though one or more of them stop working. In this type of applications low cost robots are used, with low processing capacity, which is why a simple movement strategy is necessary. It is also ideal in this type of applications the use of abstraction tools that facilitate the coding of tasks [15, 3].

Given the structure achieved in the experiments, our work is closely related to the temporal logic, since the system as such uses symbolism to represent and process information in terms of time [11, 12]. In many applications it is not clear how to construct a coordinate system for the robot, even in cases where these coordinates can be local. Therefore, in our experiments we have raised the level of abstraction to the point where the robot achieves its relative location in the environment by recognizing specific signals in a region, that is, by identifying a region [19, 17]. The navigation is made from region to region according to the local readings of the environment, and following an ergodic dynamic [18].

The experiments whose model is sought to identify in this work correspond to tasks of robot grouping [19]. In these tasks the objective is that the robot, under certain conditions, determines that a specific area of the environment is suitable for it, and therefore it must remain in the area [20]. This behavior is similar to many groups of animals. In particular, we use some rules of behavior inspired by bacteria [14].

The paper is organized as follows. In Section 2 presents a description of the problem and its mathematical formulation. Section 3 describes the strategies used to estimate the model. Section 4 introduces some results obtained with the proposed model. Finally, conclusion and discussion are presented in section 5.

2 Problem statement

A collection of n robots (numbered 1 to n) is placed into a compact, connected planar workspace $W \subset \mathbb{R}^2$. Let ∂W denote the boundary of W. There is in W a set of obstacles O, that make up areas inaccessible to robots, and whose borders are part of ∂W. The free space E corresponds to the navigation area of the robots, and is equal to W - O - ∂W.

E is decomposed into connected cells, or regions, by means of virtual gates [5]. A virtual gate connects two points of ∂W, so that the regions that are created are small in comparison to W, but large in front of the robots. In addition, each region is bounded by two virtual gates of different type and ∂W. This last condition guarantees two different points of input/output of the regions.

The virtual gate may become part of E or ∂W depending on the robot's local readings and navigation criteria. A series of events detectable by robots determine the behavior of the virtual gate as follows: Let τ the curve defining the virtual gate in W, and assuming two possible interpretations of robot local readings, [0, 1], then the behavior of the curve for the robot is $\tau(0) \in E$ and $\tau(1) \in \partial W$.

Each region may contain the n robots without organizational problems. Robots are modeled as points, regardless of geometry. In this sense, its kinematics is not important for the system architecture.

The particular equations of motion for each robot is unimportant. 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 a wild, uncontrollable way, but the trajectory satisfies the following high-level property: For any region it is assumed that the robot moves on a trajectory that causes it to strike every open interval in ∂W infinitely often, with non-zero, non-tangential velocities. A body that satisfies this property is called wild [5].

This type of motion is a specific case of ergodic system. However, in the ergodic systems the particle moves at a constant speed and bounces off the borders under the laws of reflection. In our prototypes the robots move at constant speed, but they bounce at a random angle because they are not able to measure the angle of incidence. This is the definition of wild motion.

3 Materials and Methods

Our experiments with robots seek to optimize the performance (reduce time) of grouping algorithms by including bio-inspired rules of behavior. The selected grouping algorithm was a fuzzy sliding scheme of multiagent formation control [16]. The grouping algorithm was implemented by designing a potential field between robots (a potential field equivalent to local communication between robots), and controlling the relative position of robots. The grouping was achieved by defining a constant radius between the robots of value equal to the circle that circumscribes them, i.e., $\frac{0.27m}{2}$.

Bio-inspired rules are taken from bacterial quorum sensing (QS) behavior. In short, when the population of individuals exceeds a certain size in an area, individuals emit a signal to attract the other individuals of the population [14]. In our experiments the threshold of QS is $T = 5$, which means that when there are five or more individuals in a region, the signal of attraction is emitted.

Laboratory experiments were conducted with small differential wheeled robot 27 cm $\times$ 27 cm. We use up to five of these robots in a 6 m $\times$ 6 m square navigation environment with three obstacles. The experiments were replicated in Player/Stage in order to generalize the behavior to a greater number of robots. Population size was increased to facilitate QS. However, the values remain close to those reported in [16] in order to ensure consistency in the results.

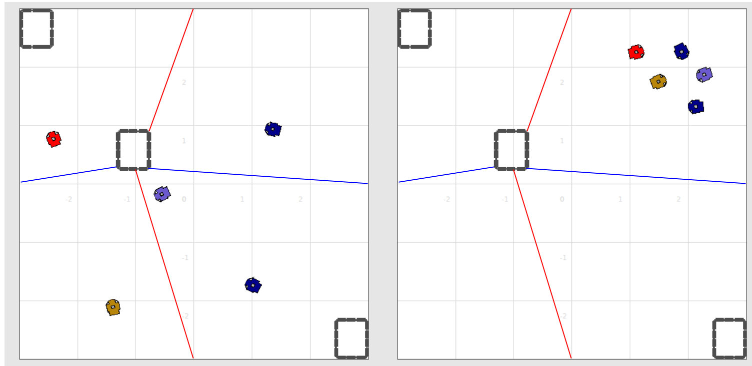

Figure 2: Navigation environment, regions, virtual gates and robots

This work presents the model of behavior on a single environment (a unique configuration of obstacles) varying the number of robots. The free space of the navigation environment was divided into four regions by four virtual colored gates (red and blue, Fig. 2). These gates are colored tapes placed on the floor that can be detected by the robot using a color sensor. Depending on the control policies, the robot can interpret these gates as open or closed. Robots detect the boundaries of the environment and obstacles by means of impact sensors. When the robot detects an environment boundary, an obstacle or a closed virtual gate, it stops, rotates a random angle, and advances again at constant speed. The QS signal is transmitted via bluetooth.

The objective of the task is to group all the robots in the upper right region. We perform 50 simulations for each number of robots (from one robot to 20 robots), always in the same environment. In each case we apply to the results an analysis of variance (ANOVA) to analyze the differences between the mean values. Since the ANOVA test provides a statistical test of whether or not the means of several groups are equal, this analysis reveals whether really the inclusion of QS algorithm contributes to reducing the total time of the task, and if the number of robots affects the total time.

4 Results and Discussion

Obtaining the model is analyzed as a regression problem. Throughout the tests with the robots they produce a sequence of time data that must allow to estimate the total time of the task for any number of robots. The data sequence is shown in Fig. 3. The figure shows an increasing trend indicating some type of memory at the output. Therefore, we apply a recurrent neuronal network type LSTM to develop the behavioral model.

LSTM networks use memory blocks instead of neurons. The blocks have gates that handle both the state of the block and its output. These memory

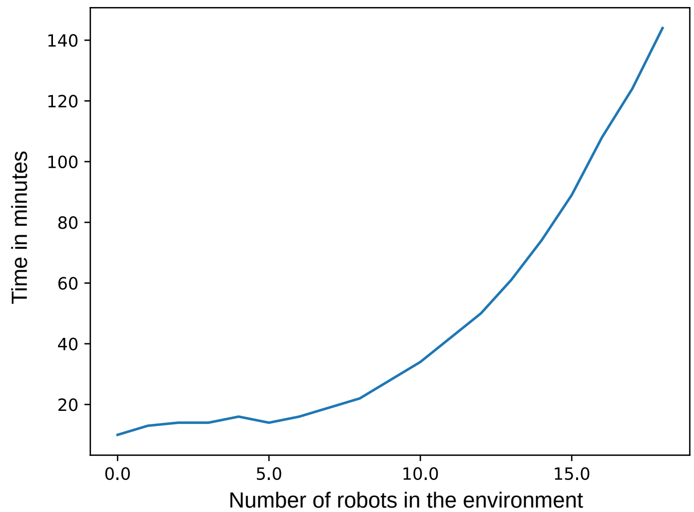

Figure 3: Plot of the total time task dataset

blocks are connected through the layers. This block structure provides higher performance than the classical neuron, and adds short-term memory to the system. The memory block receives an input sequence, and each gate inside uses sigmoid activation units to control its triggering. There are three types of gates in one unit: Forget Gate, Input Gate and Output Gate. Each unit is like a small state machine in which the gates have weights that are defined during training.

The performance of the model is evaluated using cross validation. To do this we separated the dataset in an orderly manner, creating a training set and a test set. For training we use 70% of the data, and we use the rest to test the model. The network has a visible layer with one input, a hidden layer with eight LSTM blocks or neurons, and an output layer that makes a single value prediction. The default sigmoid activation function is used for the LSTM blocks. The network is trained for 100 epochs and a batch size of one is used. The model fit code was written in Python using Keras. We setup a SciPy work environment with Pandas support. Models were evaluated using Keras 2.0.5, TensorFlow 1.2.0 and scikit-learn 0.18.2.

The results can be seen in Fig. 4. In the figure the original dataset is represented in blue, the predictions for the training dataset in green, and the predictions on the unseen test dataset in red.

We can see that the model has an average error of about 4.3 minutes on the training dataset, and about 24.4 minutes on the test dataset. These values are calculated on the same scale and dimensions of the dataset, and at full scale of the time correspond to an average error of 2.2% and 12.2%.

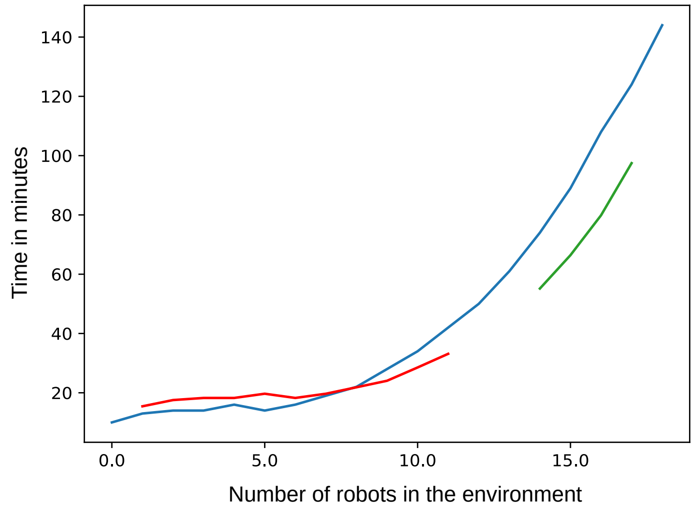

Figure 4: Plot of the total time dataset and model predictions

5 Conclusions

This paper presents a model for the estimation of the navigation time of a swarm of autonomous robots designed as finite automatons. This model constitutes a tool for the design of navigation tasks with these robots, which are characterized by a very low processing capacity, low cost and high reliability. The robots navigate the environment following the behavior of a billard, which guarantees that in finite time the robot will travel throughout a bounded region. The navigation area is segmented into regions by means of virtual gates, which in turn serve to indicate a sequence of navigation to the robots. The model for the time was identified using an LSTM network. This neuronal structure was selected by observing the importance of the previous events in the behavior. The network structure was designed with an input layer, a hidden layer with eight LSTM blocks or neurons, and an output layer. The performance of the models are calculated by evaluating the average error on the training dataset, and the average error on the test dataset. According to the results, the model predict the behavior of the signals faithfully, since in the worst case the error does not exceed 13%. Consequently, this research will continue to generate models for variants of the navigation environment, establishing baseline behavior and designing metrics for comparing similarity between environments.

Acknowledgements. This work was supported by District University Francisco José de Caldas. The views expressed in this paper are not necessarily endorsed by District University. The authors thank the research group ARMOS for the evaluation carried out on prototypes of ideas and strategies.

References

[1] R. Alur, T. A. Henzinger, G. Lafferriere and G. J. Pappas, Discrete abstractions of hybrid systems, *Proceedings of the IEEE*, **88** (2002), no. 7, 971-984. https://doi.org/10.1109/5.871304

[2] K. J. Åström, Event based control, *Analysis and Design of Nonlinear Control Systems*, in Honor of Alberto Isidori, Springer Verlag, 2008, 127-147. https://doi.org/10.1007/978-3-540-74358-3_9

[3] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins and G. J. Pappas, Symbolic planning and control of robot motion [Grand challenges of robotics], *IEEE Robotics and Automation Magazine*, **14** (2007), no. 1, 61-70. https://doi.org/10.1109/mra.2007.339624

[4] F. Bernardini, M. Gheorghe, F. J. Romero-Campero and N. Walkinshaw, A hybrid approach to modeling biological systems, Chapter in *Membrane Computing*, G. Eleftherakis, P. Kefalas, G. Paun, G. Rozenberg and A. Salomaa, eds., Lecture Notes in Computer Science, Vol. 4860, Springer Berlin Heidelberg, 2007, 138-159.
https://doi.org/10.1007/978-3-540-77312-2_9

[5] L. Bobadilla, F. H. Martinez, E. Gobst, K. Gossman and S. M. LaValle, Controlling wild mobile robots using virtual gates and discrete transitions, *2012 American Control Conference (ACC)*, (2012), 743-749.
https://doi.org/10.1109/acc.2012.6315569

[6] T. Bretl, Control of many agents using few instructions, *Proc. Robotics: Sci. Syst. Conf.*, (2007). https://doi.org/10.15607/rss.2007.iii.027

[7] P. Fraigniaud, D. Ilcinkas, G. Peer, A. Pelc and D. Peleg, Graph exploration by a finite automaton, *Theoretical Computer Science*, **345** (2005), no. 2-3, 331-344. https://doi.org/10.1016/j.tcs.2005.07.014

[8] R. Grandi, R. Falconi and C. Melchiorri, Coordination and control of autonomous mobile robot groups using a hybrid technique based on particle swarm optimization and consensus, *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, (2013), 1514-1519.
https://doi.org/10.1109/robio.2013.6739681

[9] C. Guestrin, A. Krause and A. P. Singh, Near-optimal sensor placements in Gaussian processes, *Proceedings of the 22nd International Conference on Machine Learning-ICML '05*, (2005), 265-272.
https://doi.org/10.1145/1102351.1102385

[10] T. Itami, Transporting objects only by collision with group robots, *2012 17th International Conference on Methods and Models in Automation and Robotics (MMAR)*, (2012), 639-644.
https://doi.org/10.1109/mmar.2012.6347810

[11] M. Kloetzer and C. Belta, Temporal logic planning and control of robotic swarms by hierarchical abstractions, *IEEE Transactions on Robotics*, **23** (2007), no. 2, 320-330. https://doi.org/10.1109/tro.2006.889492

[12] H. Kress-Gazit, G. Fainekos and G. Pappas, Temporal logic motion planning for mobile robots, *Proceedings IEEE International Conference on Robotics and Automation*, (2005), 2020-2025.
https://doi.org/10.1109/robot.2005.1570410

[13] G. Lee, Y. Sato and N. Chong, Decentralized task-oriented local group generation for robot swarms, *8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2011)*, (2011), 174-179.
https://doi.org/10.1109/urai.2011.6145955

[14] F. H. Martínez and J. A. Delgado, Hardware emulation of bacterial quorum sensing, Chapter in *Advanced Intelligent Computing Theories and Applications*, Lecture Notes in Computer Science, Vol. 6215, Springer-Verlag Berlin Heidelberg, 2010, 329-336.
https://doi.org/10.1007/978-3-642-14922-1_41

[15] S. Nakano, X. Zhou and T. Nishizeki, Edge-coloring algorithms, Chapter in *Computer Science Today*, Springer-Verlag, 1995, 172-183.
https://doi.org/10.1007/bfb0015243

[16] B. Ranjbar-Sahraei, F. Shabaninia, A. Nemati and S. Stan, A novel robust decentralizde adaptive fuzzy control for swarm formation of multiagent systems, *IEEE Transactions on Industrial Electronics*, **59** (2012), no. 8, 3124-3134. https://doi.org/10.1109/tie.2012.2183831

[17] A. Rendón, Evaluación de estrategia de navegación autónoma basada en comportamiento reactivo para plataformas robóticas móviles, *Tekhnê*, **12** (2015), no. 2, 75-82.

[18] D. A. Shell, C. V. Jones and M. J. Mataric, Ergodic dynamics by design: A route to predictable multirobot systems, Chapter in *Multi-Robot Systems: From Swarms to Intelligent Automata*, Springer, 2005, 291-297.
https://doi.org/10.1007/1-4020-3389-3_26

[19] K. Yoshida, H. Fukushima, K. Kon and F. Matsuno, Control of a group of mobile robots based on formation abstraction and decentralized locational

optimization, *IEEE Transactions on Robotics*, **30** (2014), no. 3, 550-565. https://doi.org/10.1109/tro.2013.2293836

[20] H. Zhang, Z. Zhao and Z. Lin, Experimental verification of a multi-robot distributed control algorithm with containment and group dispersion behaviors: the case of dynamic leaders, *IEEE/CAA Journal of Automatica Sinica*, **1** (2014), no. 1, 54-60. https://doi.org/10.1109/jas.2014.7004620

Received: September 28, 2017; Published: November 26, 2017