Biased Random Walk For Controlling a Mobile Robot

Fabio DallaLibera[*]　Shuhei Ikemoto[†]　Takashi Minato[‡]　Yutaka Nakamura[†]
Hiroshi Ishiguro[† ‡]　Emanuele Menegatti[*]

Abstract

Some bacteria like Escherichia Coli present a movement that can be modeled as a biased random walk. Biased random walk can be used for artificial creatures as a very simple and robust control policy for tasks like goal reaching. In this paper we show how a very simple control law is able to guide a simulated mobile robot equipped with an omni-directional camera toward a target without any knowledge of the actuators or of the camera. Experiments to validate the robustness of the approach show that the robot is able to reach the target having sustained several damages, namely the reduction of the size or uncontrollability of one wheel, rotation of the axle of one wheel and obscuration of 20% of the camera image. We then show experimentally that the best behavior is obtained using a bias which is roughly proportional to the random walk step, with a coefficient dependent on the physical structure of the robot, on its actuators and on and its sensors after the damage.

1 Introduction

In challenging environments like forested paths [7] or planetary explorations [6] robustness is an essential feature. Often simple living beings present a highly adaptive and robust behavior despite their structural simplicity. In particular bacteria are able to sense the concentration of nutrients and direct their movements toward the food molecules while escaping from poisoning substances, a process called chemotaxis, without any complex planning strategy or fault detection system. For instance Escherichia Coli (in the following referred as E. Coli) utilizes a biased random walk for its movement [1].

This bacterium has only two ways of moving, rotating clockwise or counter-clockwise. When it rotates counter-clockwise the rotation aligns its flagella into a single rotating bundle and it swims in a straight line. Conversely clockwise rotations break the flagella bundle apart and the bacterium tumbles in place and changes its direction randomly. The bacterium cannot therefore choose the direction of its movement and proceeds alternating clockwise and counterclockwise rotations. In absence of chemical gradients the length of the straight line paths (counter-clockwise rotations) is independent of the direction, and the bacterium essentially performs a random walk. In case of a positive gradient of attractants (like food) E. Coli instead reduces the tumbling frequency, i.e. proceeds in the same direction for a longer time biasing therefore the overall movement toward increasing concentrations of the attractant.

The same type of behavior has been applied for the control of a mobile robot [2], showing that biased random walks can be a valid approach for the navigation to sources using gradient methods. In particular the authors demonstrate that while gradient descent is faster for tracking a single source, the biased random walk performs better in the presence of multiple and dissipative sources and noisy sensors and actuators. The randomness of the algorithm also prevents the robot from ending up in local minima.

While for E. Coli and for its artificial counterpart [2] the hardware already provides two basic movements (proceed straight and change direction randomly) and the biased random walk is performed at the behavior level, in our work a biased random walk is executed directly in the *motor command space*, i.e. the behaviors themselves are determined online through the random walk. This gives great robustness in case of hardware failures since new behaviors that exploit the current hardware behavior are found online by biased random walk.

In other terms using the approach presented in [2] if a component failure causes the deterministic behavior, i.e. move forward, not to cause the expected movement, the task will not be achieved. For instance if due to an encoder problem a motor starts to rotate in the opposite direction and the robot turns around itself instead of going forward then the target will never be reached. With our approach, instead, the robot will explore new motor commands until it finds that rotating the motors in the opposite direction the distance from the target can be decreased. In general, performing a random walk in the motor command space allows to determine runtime how to exploit the dynamic of the hardware (that can change due to hardware failures) whose degrees of freedom can be hidden to the controller by a more symbolic, high level behavior representation.

More concretely in our experiments we assume we have a wheeled robot equipped with a sensor that tells us only whether the quantity we want to maximize, e.g. the presence of a chemical, is increasing

[*] Padova University
[†] Osaka University
[‡] Asada Project, ERATO, JST

or decreasing. We then suppose the controller to be required to provide as its output the angular velocity of each motor. We show that a biased random walk in the space of the actuator signals is able to drive the robot toward the source, even without any knowledge of the robot structure, e.g. with no information on the location or orientation of the wheels. To determine how to best exploit biased random walk for direct motor control we then study the effect of changing the noise and the bias amplitudes. Results show that in our settings the performance of the behavior is determined essentialy solely by the ratio of the two quantities, with an optimal ratio value dependent on the hardware state.

Section 2 reports in detail our control algorithm, sections 3 and 4 describe out experimental setup and results and section 5 illustrates future works.

2 Control Algorithm

As stated in the introduction our approach takes inspiration from the chemotaxis of E. Coli, but while this bacterium and the robot presented in [2] have basic behaviors already implemented at a low level we aim at having the system determine on-line these motion primitives too. We therefore control the robot with a biased random walk directly at the hardware level, in our case providing directly the velocities of the wheels. A big advantage of this approach is that it can recover from unexpected and even quite serious damages in the robot hardware, giving the system a high robustness without the need of introducing any self modeling or damage identification.

The principle underlying the behavior of E. Coli can be interpreted under the very general framework of biological fluctuations [3, 8]. Expressly assuming to have a continuous time system we can model it by the equation

$$\dot{u} = \alpha A(x) f(u) + \beta\eta. \tag{1}$$

where $u \in \mathbb{R}^m$ is the control signal, $x \in \mathbb{R}^n$ is the state, $f : \mathbb{R}^m \to \mathbb{R}^m$ is a deterministic function of the current control, η is a random variable, α and β are two scaling coefficients and $A : \mathbb{R}^n \to \mathbb{R}$ is a function of the state, called "activity", that indicates the fitness, or "quality" of a particular state. Intuitively when the state is getting better the value of $A(x)$ increases and control becomes mainly deterministic, while when the conditions worsen the control becomes more and more stochastic. As a practical example in the E. Coli case increasing food concentrations reduces the tumbles (random direction changes).

This simple, biologically inspired framework revealed to be very robust and perform well in many applications, for instance for routing in overlay networks [4], robot navigation [5] or control of pneumatic actuated robotic arms [3].

In our case the control signal u corresponds to the angular velocity of each motor and the state x corresponds to the information coming from our sensors. In particular our robot is equipped with an hyperbolic mirror omni-directional camera, and x is the number of pixels whose color is similar to the color of the target (see section 3 for the details).

To simplify as much as possible we decided to set $f(u)$ as follows:

$$f(u) = \frac{u}{\|u\|} \tag{2}$$

i.e. we maintain only the direction of u, and

$$A(x) = sgn(\frac{dx}{dt}) \tag{3}$$

where sgn is the sign function.

Concretely we simply apply a bias in the current direction if we are following an increasing gradient, and bias in the opposite direction if the values of the activity is decreasing and no bias in case of a constant activity.

Good values for α and β were determined experimentally. In particular we will show in section 4 that for a given value of α the optimal value of β (in terms of average performance maximization) is roughly proportional to α.

3 Experimental setup

Using ODE* we simulated a mobile robot equipped with three spherical wheels. The two front wheels are directly actuated by two independent motors whose maximum velocity is 0.5 rad/s while the rear wheel is free to rotate in any direction. The task is to reach a red semisphere of radius 4 m placed at a distance of 30m. The robot is equipped with an omni-directional camera and the value of x fed to the controller is the number of red pixels in the image, determined by a filter that given the RGB components of each pixels counts the pixels whose R component is more than double the maximum of the G and B component values.

The controller receives information on the red pixels with a sampling frequency of 0.2Hz and provides a 2 dimensional velocity command u. We chose to employ such a low sampling frequency to validate the robustness of the method even in case of low cost hardware with very poor performances. The controller implements the discrete time equivalent of equation 1, expressly

$$u_{t+1} = u_t + \alpha A(x) \frac{u_t}{\|u_t\|} + \beta\eta \tag{4}$$

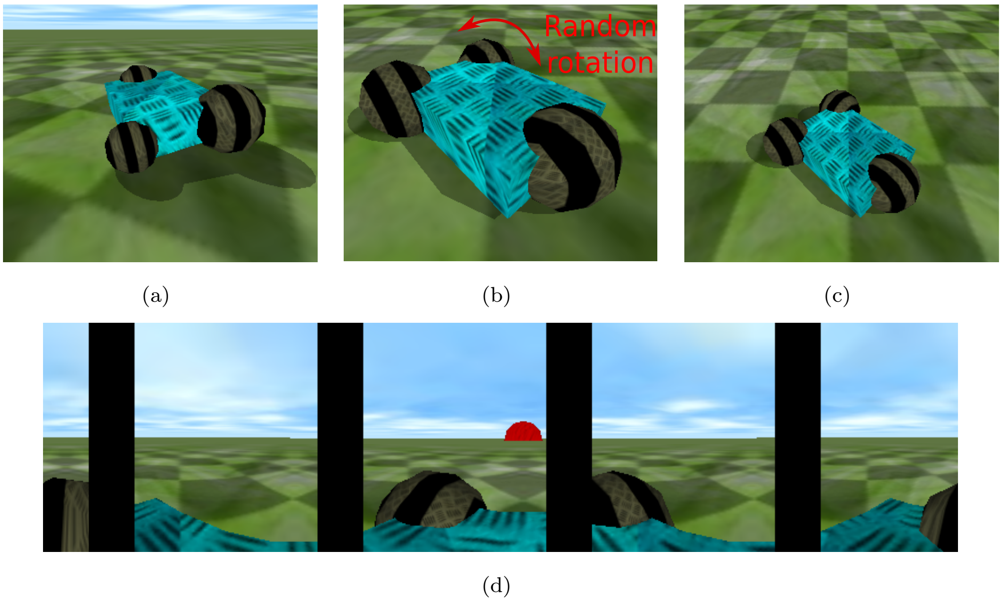
where

$$A(x) = sgn(x_t - x_{t-1}) \tag{5}$$

We simulated four types of damage (see Figure 1):

1. the right wheel size is reduced to two thirds of its normal size
2. the right wheel becomes uncontrollable, and its movement is completely random
3. the right wheel rotation axis direction is turned 90 degrees along the Z axis and becomes parallel to the longitudinal axis, i.e. the rotation of the wheel instead pushing the robot forward and backward pushes the robot towards sideways

* For details see `http://www.ode.org`.



(a) (b) (c)

(d)

Fig.1 The four damages simulated. (a) Reduced size of one wheel. (b) Uncontrollability of one wheel (c) Rotation of the rotation axis of one wheel (note the direction of the stripes) (d) Obscuration of part of the image.

4. 20% of the camera image becomes obscurated

We took $\eta \sim \mathcal{N}(0,1)$ as a Gaussian variable of variance one and studied the behavior for several values of α and β. In particular for each condition (no damage or one of the damages listed) we determined the time spent contacting with the goal over 20000 seconds. Each condition is simulated for 128 different positions of the target, in particular assuming the robot's chassis is placed at (0,0) we set the target in each of the positions as $(R \cdot cos(\theta_i), R \cdot sin(\theta_i))$, $\theta_i = \frac{i \cdot 2\pi}{128}$, $i = \{0, \ldots, 127\}$ where $R = 30$ m.

4 Results

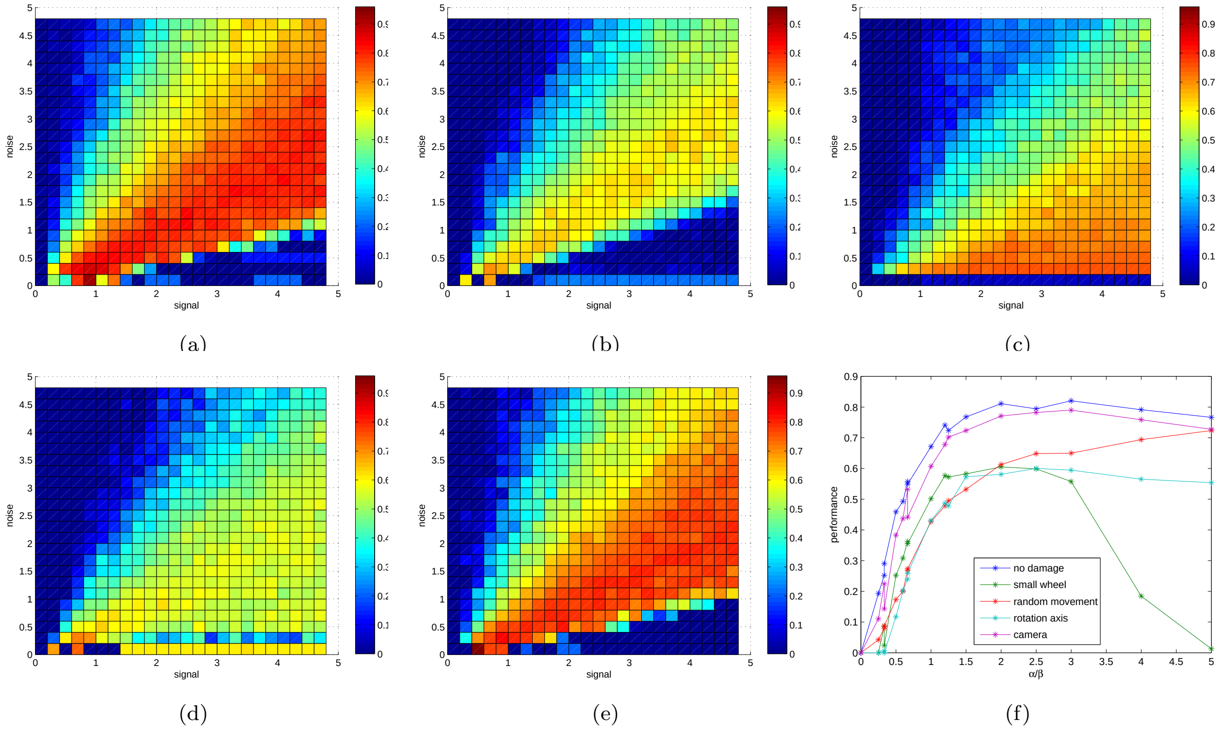
Although the motors maximum speed is 0.5 rad/s in all experiments the best results were obtained for values of α and β higher than 1. For high values of α and β the system essentially selects between four possible behaviors which move the robot forward, backward or make it rotate or slide (in the case of the wheel axis damage) with the maximum speed. Always using the maximum speed means that when the robot takes a wrong action this brings the robot away from the target to the maximum extent, but if the robot succeeds to reach the target, trivially it must mean that more than 50% of the movements bring the robot closer to the destination. In such setting having the robot to move at the maximum speed ensures the best performances.

Figure 2 depicts the results for the undamaged robot and for the four damages previously listed. The x and y axis indicate the values of α and β respectively, while the color represents the performance, in terms of ratio between the time spent touching the target and the total simulation time (20000 seconds). For all damages the graphs presents non zero values, i.e. the robot is able to reach the target and touch it. As expected a completely deterministic behavior ($\beta = 0$) is often not able to drive the robot to the target, since without "exploration" of motor commands done by the random part the system can just provide a single type of motor command. Similarly when $\alpha = 0$ the probability of touching the goal with a completely random movement is so low that in no experiment the robot could reach the target within the simulation time.

We can notice that the color zones are approximately triangles departing from the origin, i.e. the performance depends just on the ratio between α and β and not on their value (except the fact that they should be higher than 1 to ensure the maximum wheel speed). We can explain this by noticing that the ratio between α and β defines the behavior of the random walk, while their values define the entity of the variations and since the values of the velocity are clamped to 0.5, when α and β are high enough essentially the behavior is the same.

Figure 2(f) shows the average performance for various $\frac{\alpha}{\beta}$ ratios. We notice that for the first type of damage (reduced wheel size) a ratio close to 2 gives the best performances, while in the case of changed rotation axis the best performance is obtained with $\frac{\alpha}{\beta} \approx 2.5$. The undamaged robot and the robot with damaged camera instead performs best with $\frac{\alpha}{\beta} \approx 3$. For the uncontrollable wheel higher values for $\frac{\alpha}{\beta}$, around 5, gives the best performance. In this case probably the noise introduced by the hardware itself reduces the noise required in the control signal.

Observing Fig. 2 we notice that some damages seems easier to recover than others, in detail the performance decreases more abruptly when the size of one wheel is reduced and when the rotation axis is changed by 90 degrees than when the camera is partially obscurated or when one wheel become uncontrollable. In these cases a lower $\frac{\alpha}{\beta}$ is more beneficial,



(a) (b) (c)

(d) (e) (f)

Fig.2 Performances of the robot for various settings of α and β: (a) Without any damage, (b) Reduced size of one wheel, (c) Uncontrollability of one wheel, (d) Rotation of the rotation axis of one wheel, (e) Obscuration of part of the image, (f) plot of α/β vs. average performance.

i.e. intuitively speaking when the task is difficult the more stochastic the control is the better it is.

5 Conclusions and Future Work

In this paper we presented an approach for mobile robot navigation based on biased random walk and inspired from the movement of E. Coli chemotaxis. We showed that a random walk in the motor command space can be sufficient to drive a robot toward a goal without any knowledge of the robot structure. This ensures that even in case of hardware failure the control adapts to the new robot conditions without any need of modeling and identification of the possible failures. We validated the methodology using a simulated mobile robot equipped with an omnidirectional camera, and shown that the robot is able to reach the goal even in case of severe damages of the sensors and actuators. We then provided an experimental study on the optimal bias in the random walk. These tests suggest that the optimal bias is proportional to the noise, but the coefficient depends on the hardware. Furthermore, we saw that in this case the use of a limited number of motor primitives (moving each motor with its maximum speed and change just its rotation direction) gave good performances for goal reaching.

Future works will aim at identifying whether other types of noise are more efficient than Gaussian Noise. For instance [5] shows experimentally that Levy walk performs better that random noise in a goal reaching task similar to the one presented in this paper.

Acknowledgments

This research was supported by "Special Coodination Funds for Promoting Science and Technology: Yuragi Project" of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1) J. Adler. The sensing of chemicals by bacteria. *Scientific American*, 234:40–47, 1976.
2) Amit Dhariwal, Gaurav S. Sukhatme, and Aristides A. G. Requicha. Bacterium-inspired robots for environmental monitoring. In *2004 IEEE Intl. Conf. on Robotics and Automation (ICRA 2004)*, pages 1436–1443, New Orleans, USA, 2004.
3) Ippei Fukuyori, Yoshio Matsumoto Yutaka Nakamura, and Hiroshi Ishiguro. Flexible control mechanism for multi-dof robotic arm based on biological fluctuation. In *10th Intl. Conf. on Simulation of Adaptive Behaviour, SAB 2008*, pages 22–31, Osaka,Japan, 2008.
4) Kensji Leibnitz, Naoki Wakamiya, and Masayuki Murata. Biologically inspired self-adaptive multi-path routing in overlay networks. *Communications of the ACM*, 49:63–67, 2006.
5) Surya G. Nurzaman, Yoshio Matsumoto, Yutaka Nakamura, Satoshi Koizumi, and Hiroshi Ishiguro. Yuragi-based adaptive searching behavior in mobile robot: From bacterial chemotaxis to levy walk. In *2008 IEEE International Conference on Robotics and Biomimetics*, 2008.
6) Christian Plagemann, Dieter Fox, and Wolfram Burgard. Efficient failure detection on mobile robots using particle filters with gaussian process proposals. In *IJCAI*, pages 2185–2190, 2007.
7) Yoichi Morales Saiki, Eijiro Takeuchi, and Takashi Tsubouchi. Vehicle localization in outdoor woodland environments with sensor fault detection. In *ICRA*, pages 449–454, 2008.
8) T. Yanagida, M. Ueda, T. Murata, S. Esaki, and Y. Ishii. Brownian motion, fluctuation and life. *Biosystems*, 88:228–242, 2006.