Jst Erato Asada Project

1A1-F14

認知ロボティクス＆脳科学・神経科学とロボティクス

6月15日 コアタイム 10:15～11:00

Robot Control Inspired by Escherichia Coli Chemotaxis

大腸菌の走化性に基づいたロボット制御

○ダーラリベラ・ファビオ(パドヴァ大)
Fabio DallaLibera (Padova University)
池本 周平(阪大)
Shuhei Ikemoto (Osaka University)
港隆史(JST ERATO)
Takashi Minato (JST ERATO Asada Project)
石黒浩(JST ERATO / 阪大)
Hiroshi Ishiguro (JST ERATO Asada Project/Osaka University)
メネガッティ・エマヌエレ(パドヴァ大)
Emanuele Menegatti (Padova University)

Abstract

Living beings like bacteria search food using extremely simple strategies that reveal to be very robust. From this observation we derive an algorithm for robot control.

Most literature on the topic defines basic robot behaviors that are used to mimic bacteria movement in the physical space. Instead, this paper proposes to work directly in the motor command space, allowing the robot to determine itself the control signals to use. The strategy underlying the algorithm consists in simply repeating control signals that lead to condition improvements.

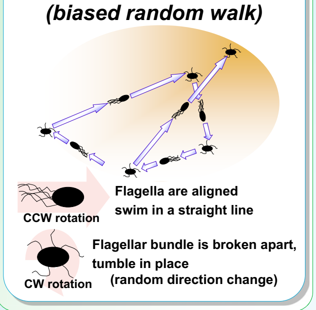
Counter intuitively, adding random perturbations to the control signal being repeated improves the performances. A model of the phenomenon is provided and an algorithm for automatic choosing the perturbation magnitude is proposed.

Results of experiments on the robustness and practical applicability of the approach are reported. In particular we show that the algorithm is able to perform robot navigation even when no information on the robot structure is available and the robot undergoes hardware damages.

Related works

Bacteria using a **very simple strategy** can reach food sources despite noise and world uncertainties [1]

Escherichia Coli chemotaxis (biased random walk)



Robots with similar behavior proposed [2]

- ✓ **Robust to noise**
- ✓ **Do not stuck in local minima**
- ✓ **Robots distribute among multiple sources**

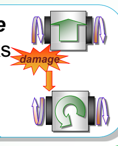
Switching between two behaviors

- **Rotate**
- **go straight**

The robot may not reach the goal due to hardware faults

Damage example

An encoder breaks
→ "go forward" becomes "spinning"



For robot fault detection and recovery very advanced techniques available[3][4]

Apply ***biased random walk*** directly in the ***command space***

- **appropriate behaviors that exploit the working hardware are found**

Control algorithm

If the conditions improved (e.g. got closer to the goal)
  keep the previous motor command
else
  choose a random motor command

$$u^i_{t+1} = \begin{cases} u^i_t + \eta^i_t R & if\ \Delta A_t \geq 0 \\ random\ selection & otherwise \end{cases}$$

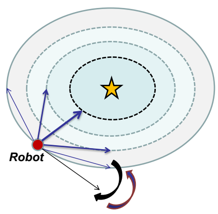
Performance improvement by noise

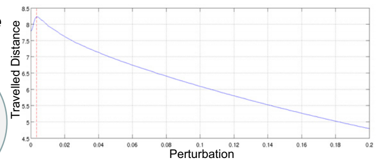
Stochastic resonance

Surprisingly, **adding random perturbations** of an opportune magnitude to the control input **improves the performance**

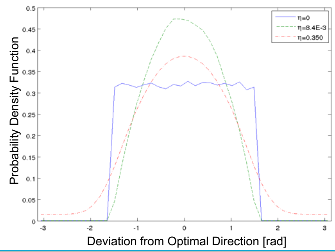
$$u^i_{t+1} = \begin{cases} u^i_t + \eta^i_t R & if\ \Delta A_t \geq 0 \\ random\ selection & otherwise \end{cases}$$

Goal reaching example



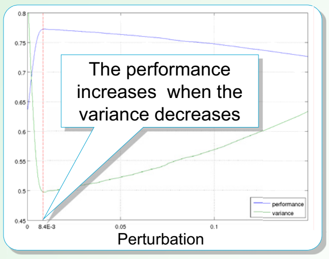


Random perturbations help to detect commands that improve the robot condition (proximity to the goal) just slightly



Automatic selection of the perturbation amplitude

- Appropriate perturbation amplitudes give good control inputs that can be used for long time
- The *control input variance* decreases when the perturbation is appropriate
- The control input variance can be used to automatically select the appropriate perturbation amplitude



Perturbation magnitude adaptation algorithm

$$\delta^i_0 = 1.1$$

$$\sigma^i_t = \frac{(u^i_t - u^i_{t-1})^2}{2}$$

$$\delta^i_{t+1} = \begin{cases} 1/\delta^i_t & if\ t\ odd \wedge \sigma^i_t \geq \sigma^i_{t-2} \\ \delta^i_t & otherwise \end{cases}$$

$$\eta^i_{t+1} = \begin{cases} \eta^i_t \delta^i_{t+1} & if\ t\ odd \\ \eta^i_t & otherwise \end{cases}$$

Performance measurements

Nonlinearities of the system

Control algorithm

$$u^i_{t+1} = \begin{cases} u^i_t + \eta^i_t R & if\ \Delta A_t \geq 0 \\ random\ selection & otherwise \end{cases}$$

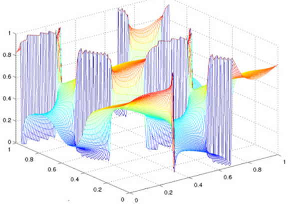
Condition evaluation

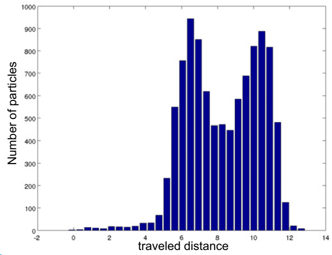
$$\Delta A_t = \|x_{t-1}\| - \|x_t\|$$

Nonlinear system dynamics

$$x^i_{t+1} = x^i_t + s \cdot f^i(u_t)$$

$$f^i(u) = \frac{1}{\pi} \arctan\left(\frac{(sin(2\pi u + \xi^i))^T Q^i sin(2\pi u + \xi^i)}{(sin(2\pi u + \zeta^i))^T P^i sin(2\pi u + \zeta^i)}\right)$$



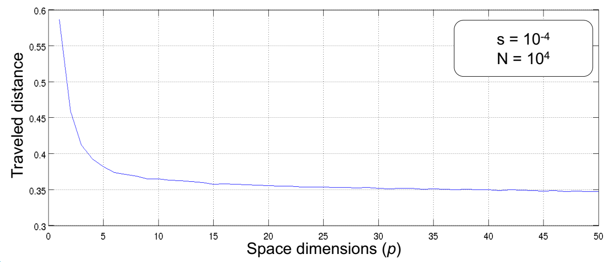


Distribution of the distance toward the goal traveled by 10^4 particles in $N = 10^4$ steps of size $s = 10^{-3}$

Dimensions of the search space

$$x_0 = [-1, 0, \ldots, 0]^T \in \mathbb{R}^p$$

$$x_{t+1} = x_t + s \cdot u_t$$

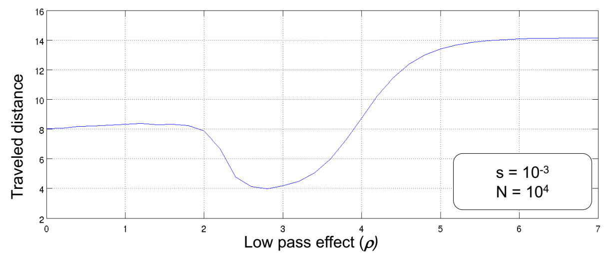


Low-pass filtering effects in the system dynamics

Low-pass system dynamics

$$v_t = (1 - 10^{-\rho}) v_{t-1} + 10^{-\rho} u_t$$

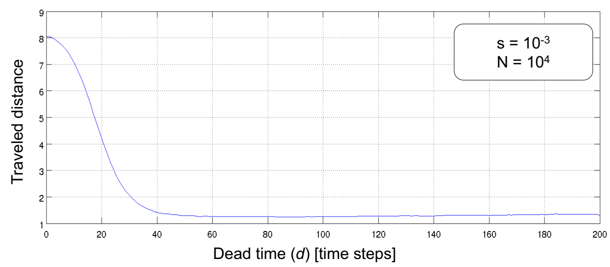
$$x_{t+1} = x_t + v_t$$



Dead time in the system dynamics

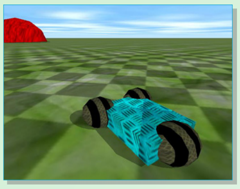
Dead time system dynamics

$$x_{t+1} = x_t + u_{t-d}$$



Experiment

Setup

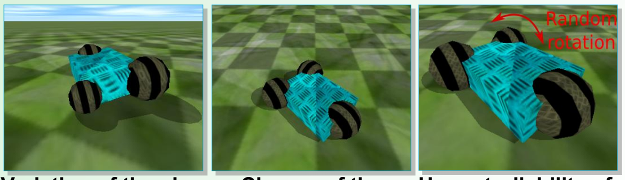


Robot: simulated mobile robot equipped with two independent wheels and an omnidirectional camera

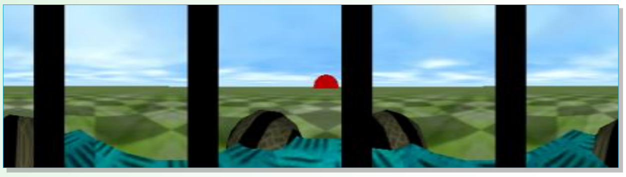
Task: reach a red hemisphere

Sensory information: number of red pixels in the camera image

Simulated four different damages



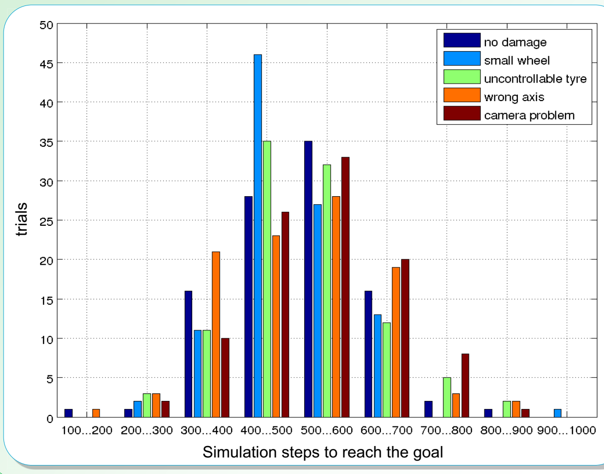
Variation of the size | **Change of the rotation axis** | **Uncontrollability of a wheel**



Obscuration of 20% of the camera information

Results

- The algorithm could lead the robot to the target without any prior information on the robot structure (axis orientation, wheel position, etc.)
- The robot was able to reach the goal for each of the tested damages



References

- [1] Adler, J. “The sensing of chemicals by bacteria”. Scientific American, vol. 234, 40–47, 1976.
- [2] Dhariwal, A., Sukhatme, G.S. and Requicha, A.A.G. “Bacterium-inspired robots for environmental monitoring”. ICRA 2004, pp. 1436–1443, 2004.
- [3] Scheutz, M. and Kramer, J. “Reflection and reasoning mechanisms for failure detection and recovery in a distributed robotic architecture for complex robots”. pp. 3699–3704. 2007.
- [4] Bongard, J., Zykov, V. and Lipson, H. “Resilient machines through continuous self-modeling”. Science, vol. 314(5802), 1118–1121, 2006.