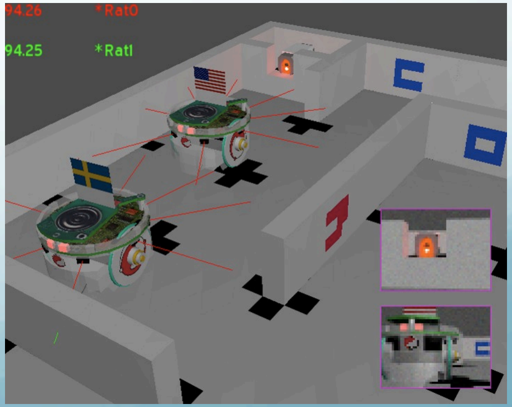
Emergence of Intelligent Behavior From A Minimalistic Stochastic Model for Robots

Sun Zhe, Micheletto Ruggero
Yokohama City University



NAVIGATING A ROBOT WITHOUT RULES, WITH A MINIMAL PURELY STOCHASTIC ALGORITHM

Contents

- **The Brain and The States Transition Matrix**
- **The E-puck and the Webots**
- **Learning algorithm**
- **The Monte Carlo method**
- **Solving the maze with "STM"**
- **Result and discussion**

AN EXAMPLE OF STM:

INPUT

A happy new year

↓

A_ _h ha ap pp py y_

_n ne ew w_ _y ye ea ar

READ TEXT FROM SOURCE

STM IS CREATED

A happy new year

↓

A_ _h ha ap pp py y_

_n ne ew w_ _y ye ea ar

	A	h	a	p	y	n	e	w	r	
A	0	0	0	0	0	0	0	0	0	1
h	0	0	1	0	0	0	0	0	0	0
a	0	0	0	$\boxed{1}$	0	0	0	0	1	0
p	0	0	0	1	1	0	0	0	0	0
y	0	0	0	0	0	0	1	0	0	1
n	0	0	0	0	0	0	1	0	0	0
e	0	0	1	0	0	0	0	1	0	0
w	0	0	0	0	0	0	0	0	0	1
r	0	0	0	0	0	0	0	0	0	0
	0	1	0	0	1	1	0	0	0	0

LEARNING STATES TRANSITIONS

RESULT OF LEARNING FROM THE STM

In a disputed election. Iran has eased during Ashura have a boost in a civilian nuclear bomb. Interviewer Diane Sawyer offered Mr. Ahmadinejad's charge the dictator," in a Nigerian Charged with Trying to prove that has until this report that received nearly $3 billion, 70 percent of the bomb killed in the local government official exploded in Indonesia," said Foreign Minister Mottaki's comments follow some 650 kilometers northeast of ships over the Red Cross Federation's Tsunami unit,

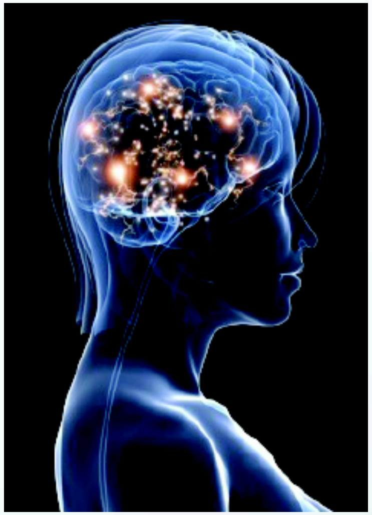
Artificially generated Text from STM. Correct and locally meaningful. No grammar rules or English syntax are used (!)

Stochastic Properties In Our Brain

- **Human behaviour** is always characterized by **a degree of randomness**.

Two properties in our brain

- 1.Very simple rules
- 2.Randomness



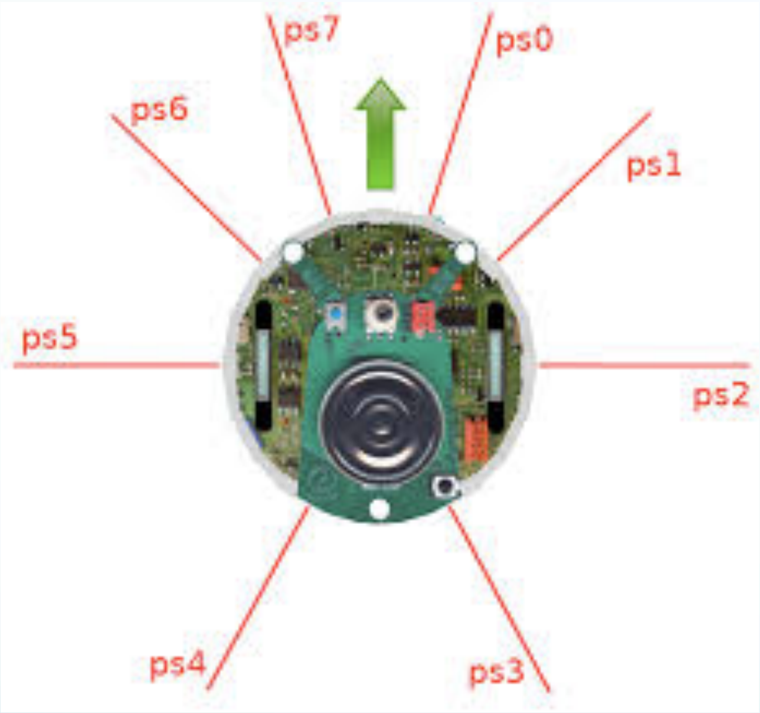
Very simple rules

Behaviour is based on simple rules

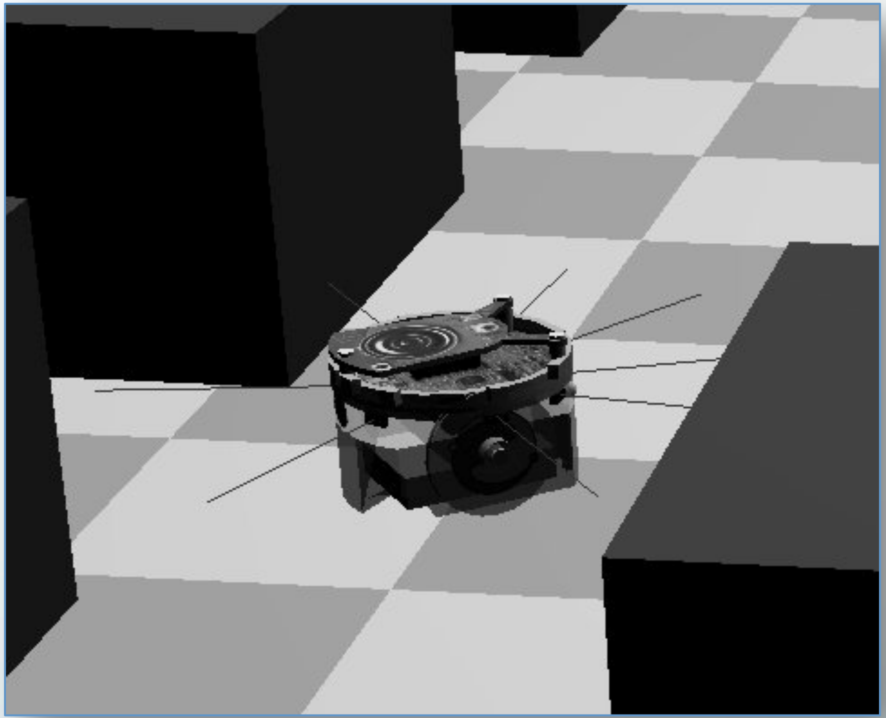
Randomness

we never repeat exactly the same movement,
nor we say exactly the same things

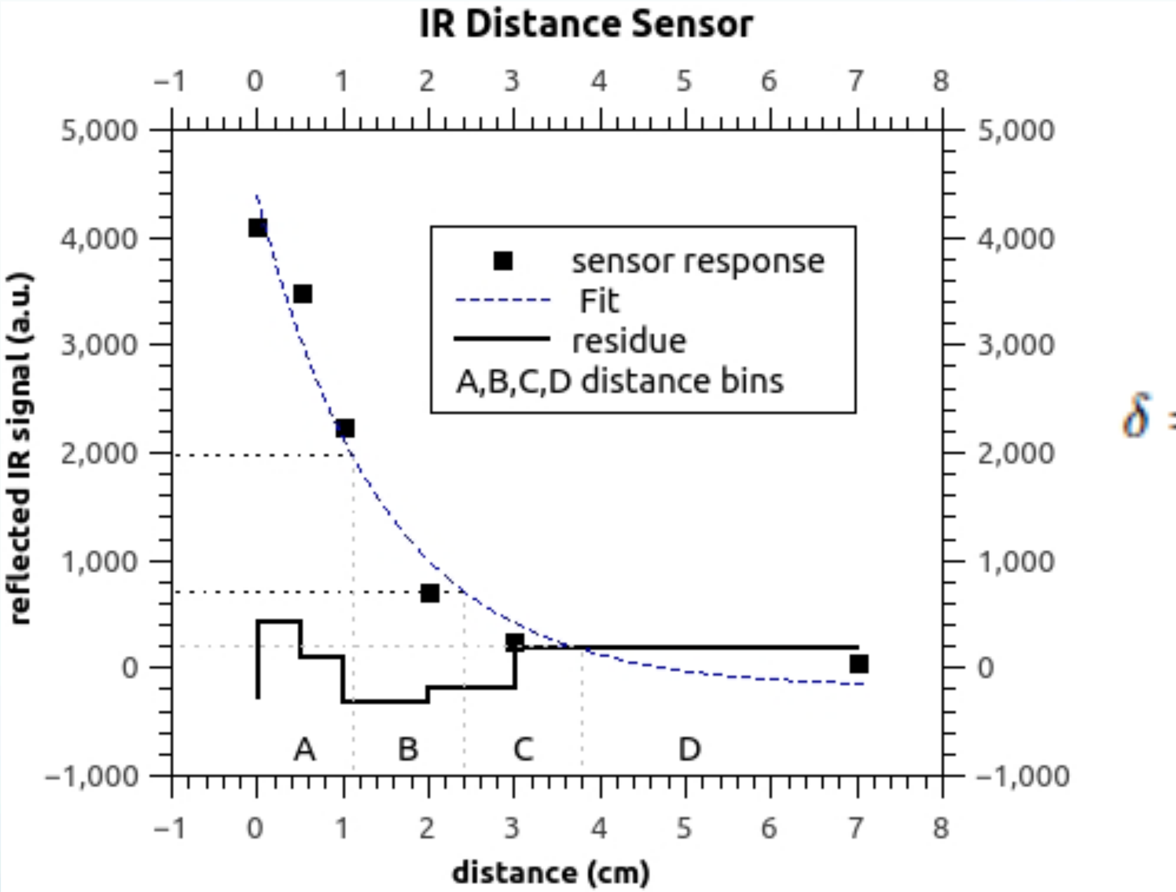
The E-puck robot and Webots



A screenshot of the real-time simulated representation of the ePuck Robot with sensors



The E-puck in webots

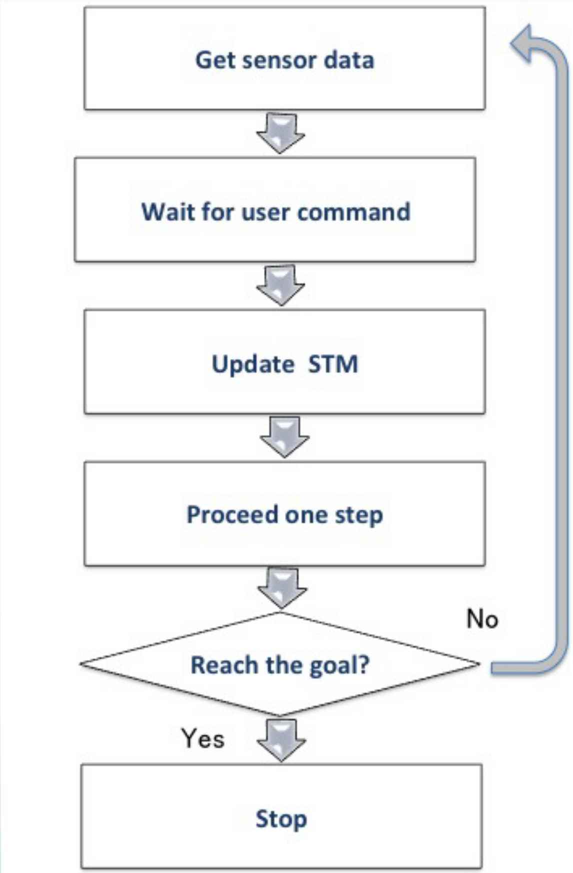


$$\delta = \left\{ \begin{array}{c} A : 0 \le s_i \le 12 \\ B : 12 \le s_i \le 24 \\ C : 24 \le s_i \le 38 \\ D : s_i > 38 \end{array} \right\}$$

The distance "STATE" vector

The IR sensors output against distance. The values are fitted against an exponential and are classified in four level bins.

Learning algorithm



a classifier operator δ that uses the four bins labelled A, B, C and D

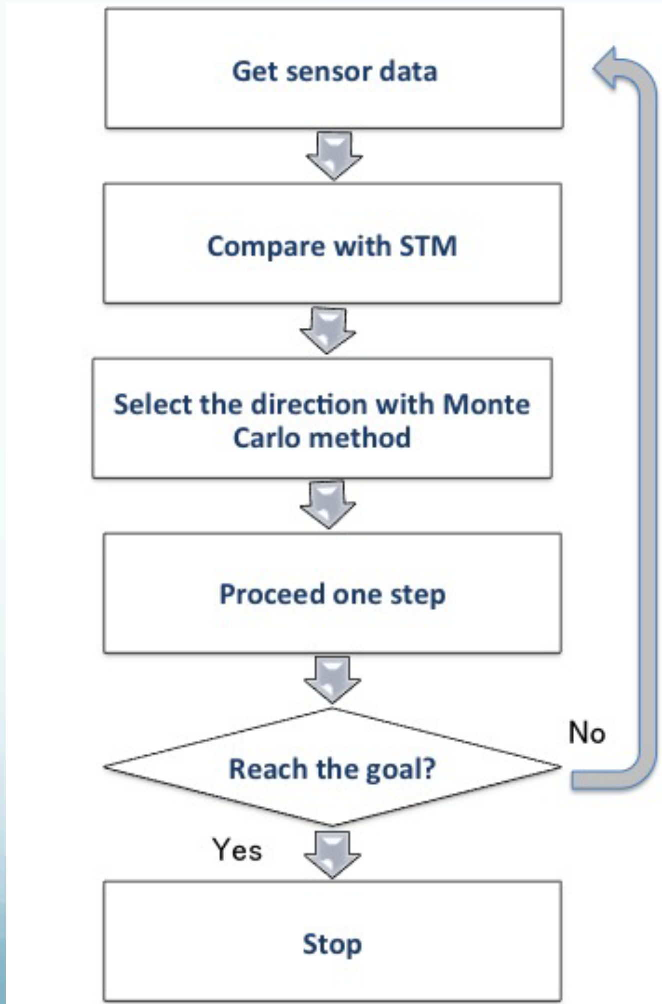
$$\delta = \left\{ \begin{array}{c} A : 0 \leq s_i \leq 12 \\ B : 12 \leq s_i \leq 24 \\ C : 24 \leq s_i \leq 38 \\ D : s_i > 38 \end{array} \right\}$$

$$\xi_n = \begin{bmatrix} \gamma_n \\ \delta \left[PS_n^* \right] \end{bmatrix} = \begin{bmatrix} \gamma_n \\ \delta_n^1 \\ \delta_n^2 \\ \delta_n^3 \\ \delta_n^4 \end{bmatrix} = \begin{bmatrix} \gamma_n \\ \Sigma_n \end{bmatrix}$$

A schematic representation of the learning phase Algorithm. The robot is guided by a user out of the maze.

δ_n^i are the resulting bins A,B,C, or D for each sensors *i* during loop *n*

The STM Driver for Running the Robot

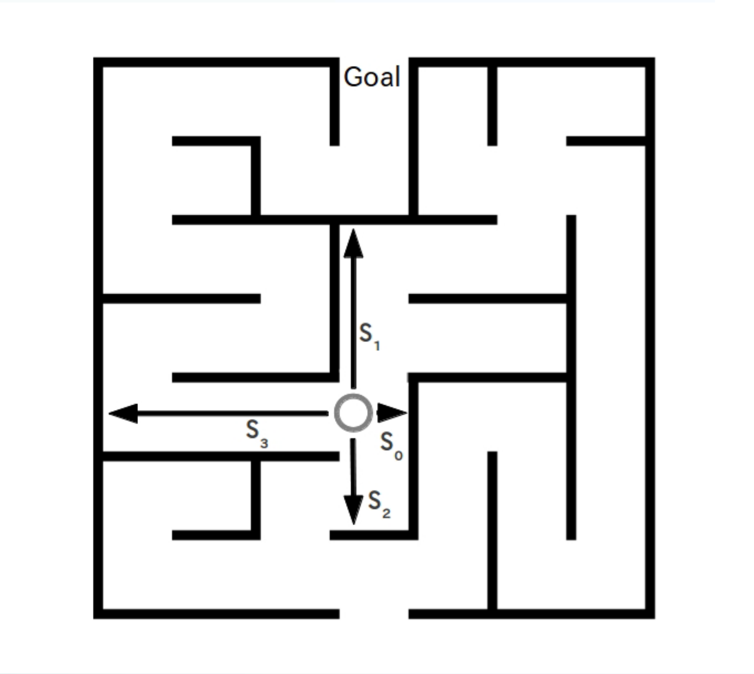


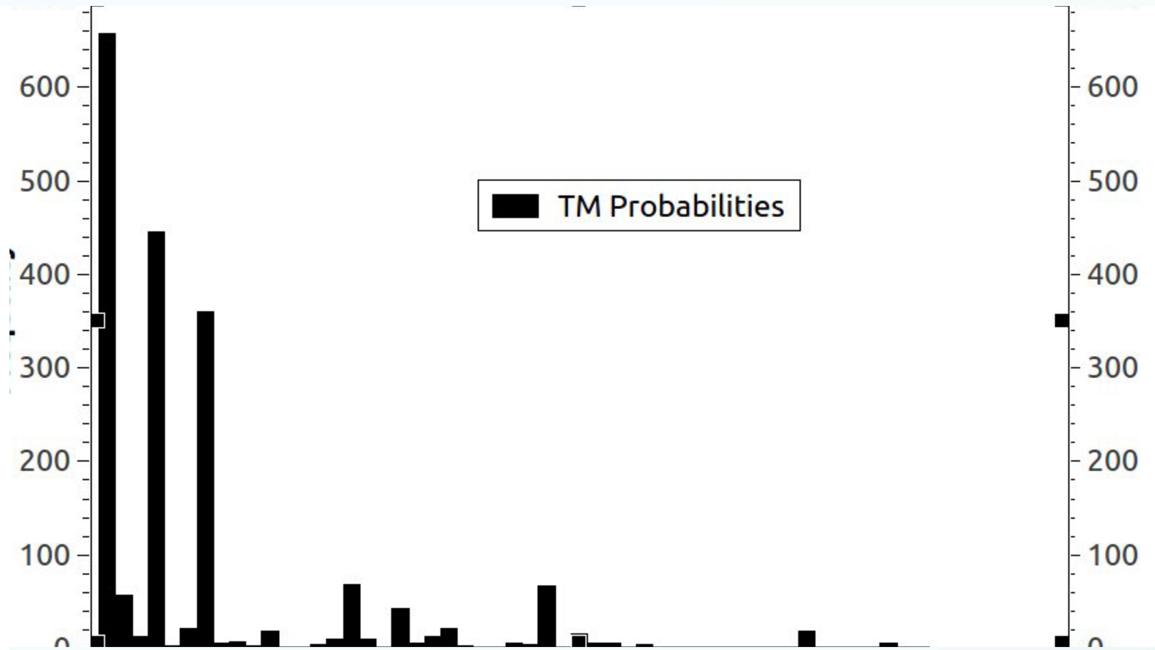
For each step the robot accesses the transition matrix STM and a Monte Carlo algorithm is used to determinethe next action.

$$S_{\sigma_n} \begin{cases} if\sigma_n \in \Sigma \rightarrow STM = \Gamma|_{\sigma_n} = \begin{Bmatrix} P_l \\ P_f \\ P_r \end{Bmatrix} \\ if\sigma_n \notin \Sigma \rightarrow STM = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix} \end{cases}$$

the second condition $\sigma_n \notin \Sigma$ is forced in order to proceed straight in the case of unknown sensory state σ_n. Once $\Gamma|_{\sigma_n}$ is calculated, a Monte Carlo method is used to choose next direction,

The Mazes and the Data in STM



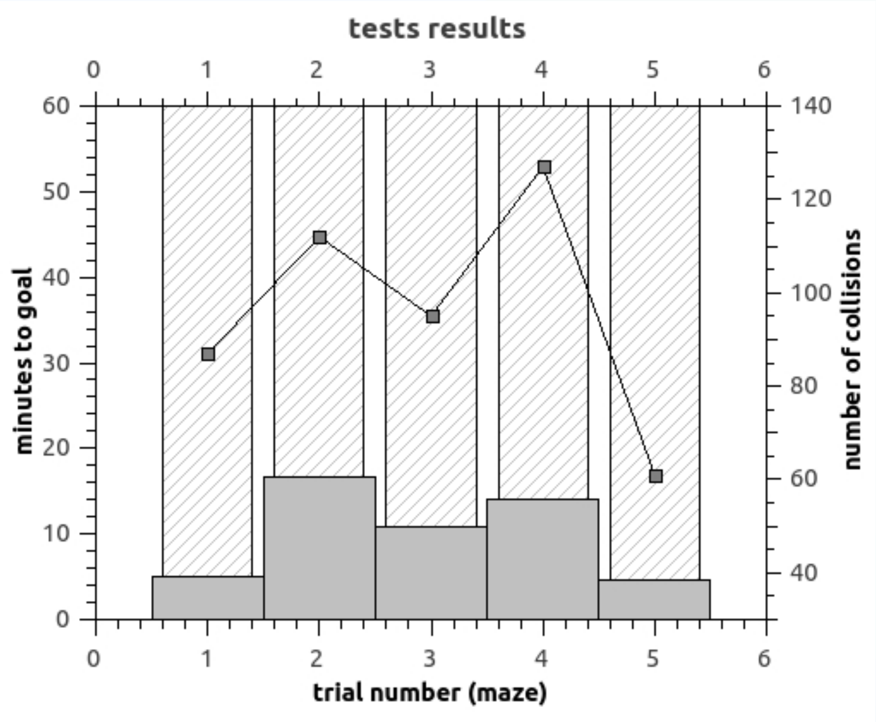


A representation of the robot in a random generated maze and its IR sensor values.

A plot of one of the Transition Matrix used in the tests

Result

Matrix	TM	Control
Test 1	5:00	>60:00
Test 2	16:49	>60:00
Test 3	10:51	>60:00
Test 4	14:02	>60:00
Test 5	4:45	>60:00

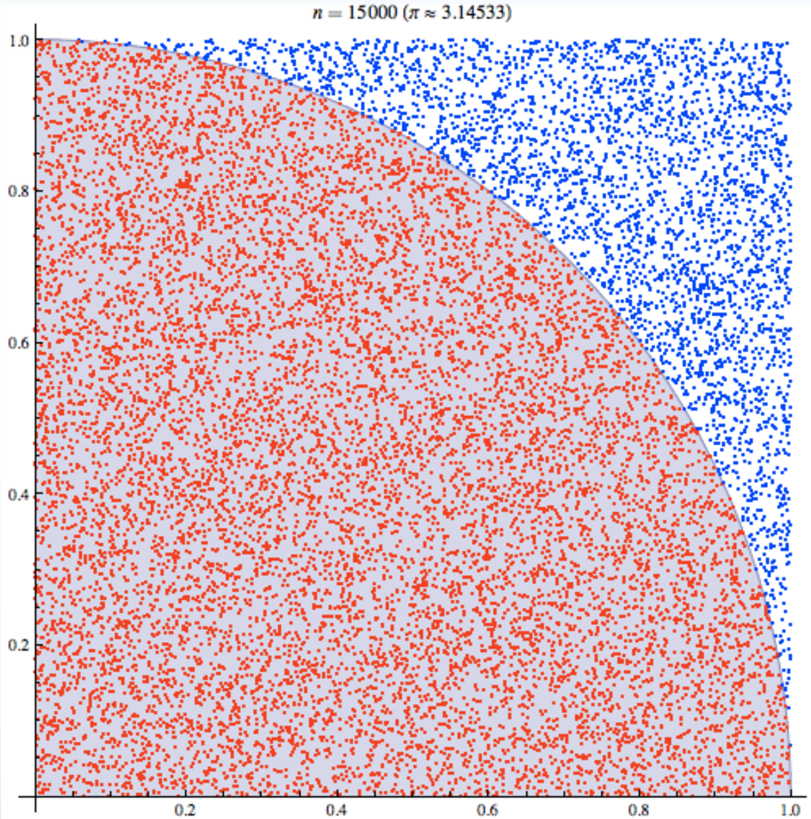


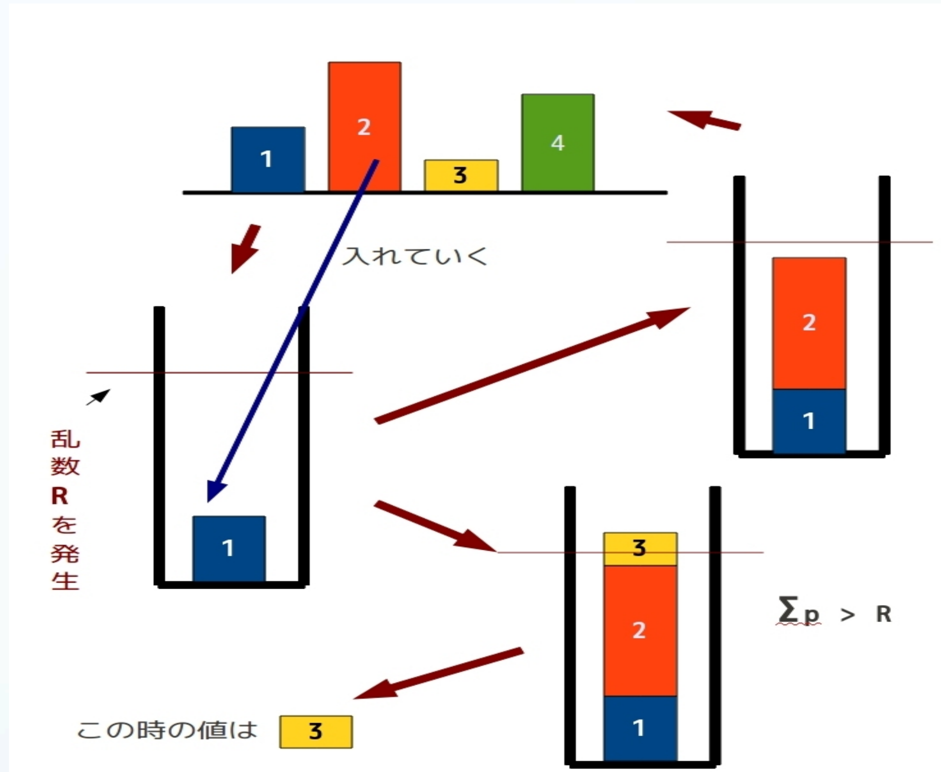
The algorithm was tested on randomly generated mazes. The robot is not aware of the position of the goal nor it uses any extra empiric knowledge to move around the maze. The navigation is done solely by the algorithm described above.

Conclusion

- the robot was able to solve the maze reaching the goal in a minimum time, this is a very interesting emerging result if we consider that the algorithm is stochastic, does not have any knowledge of the goal or its own position and that it uses a one-dimensional Markov model without memory
- We are currently setting up a new investigation in which the same mathematical approach is used to control a robotic arm

The Monte Carlo method





$$X \leqslant \sum_{k=0}^{\zeta} \Gamma|_{\sigma_n}$$

As points are randomly scattered inside the unit square, some fall within the unit circle.

the summation is done ζ times while the equation is true. The index ζ points to the next directional command $\gamma_\zeta \in \{L, F, R\}$ executed by the robot.