

Robot Learning: obtaining good results with a few experiments on the robots

Luca Iocchi

A. Cherubini, F. Giannone, P.F. Palamara, E. Menegatti, F. Dalla Libera

DIS, University of Rome “La Sapienza”, Italy

University of Padua, Italy

DIPARTIMENTO DI INFORMATICA
E SISTEMISTICA ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA

Robot Learning

- Many *robot tasks* require fine tuning of parameters in the implementation of behaviors, control actions, and strategic decisions.
- Both application of standard ML approaches and definition of new robot learning algorithm or methods
- **Challenges** in Robot Learning
 - Time and Hardware consumption
 - High noise, non-determinism
 - Real world real time requirements
 - Task complexity
 - Small amount of data (experiments) available

Robot Learning

- **Robot Learning proposes a change of focus**

From “*studying convergence properties of learning methods*”...

to “*obtaining best results with a limited (small) number of experiments*”

- **Robot Learning Tasks**

- Object/situation recognition
- Navigation optimization
- Behavior/skill learning
- Localization and world modeling
- Team behaviors (multi-robot)
- ...

- **Robot Learning Methods**

- Decision Trees
- Neural networks
- SVM
- Genetic Algorithms
- Reinforcement Learning
- ...

Robot Learning in RobCup 4LL

Year	# Teams	DT	NN/SVM	EC/RL	Other	# Pub
1998	3	-	-	-	2	2
1999	9	-	1	1	1	3
2000	12	1	1	1	2	5
2001	16	2	3	3	2	8
2002	19	2	-	3	4	9
2003	24	2	3	6	3	12
2004	24	2	1	11	5	19

Robot Learning is a winning approach!

Typical Robot Task Learning

S_1, \dots, S_m : strategies for achieving a task \mathcal{T}

$S_j = \{ B_1; B_2; \dots ; B_n \}$ (composition of behaviors)

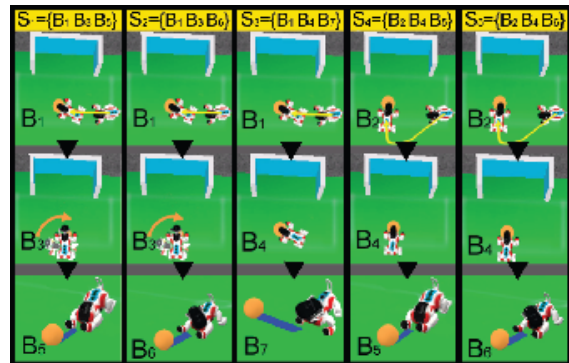
$B_i = \langle \theta_1, \dots, \theta_k \rangle$ (behavior parameters)

Example: soccer robot

B_1, B_2 : ball approaching

B_3, B_4 : ball controlling

B_5, B_6, B_7 : kick actions



First case study: Robot Task Learning

- **Problem**

- learning a complex task as a composition of different behaviors
- learning optimal parameters of the behaviors
- behavior learning and parameter learning at the same time

- **Approach**

- Extended Policy Gradient algorithm that consider *relevant* parameters and *contiguous* strategies

- **Advantages**

- fast convergence with limited (small) number of experiments
- considering different sets of parameters for the same behavior when associated to different strategies

Policy Gradient for Concurrent Behavior and Parameter Learning

Trivial use of Policy Gradient

For each strategy v

$$X_v^* \leftarrow \text{PG}(X_v^0, n_{iter})$$

$$v^* \leftarrow \text{argmax}_v F(X_v^*)$$

return $(S_{v^*}^*, X_{v^*}^*)$

An extended version of the Policy Gradient algorithm

Parameter relevance

measure the relevance of parameters wrt to a strategy

$$R(S^v, j) \rightarrow 0 \quad j \text{ not relevant for } S^v \\ \rightarrow 1 \quad \text{very relevant}$$

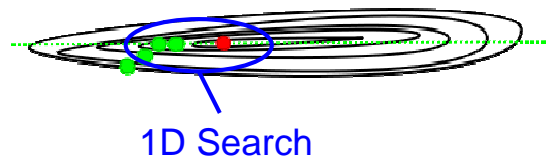
Strategy contiguity

relates solutions of two strategies

$$C(S^v, S^w) \rightarrow 0 \quad \text{different solutions} \\ \rightarrow 1 \quad \text{similar solutions}$$

Parameter Relevance: Examples

Fast convergence by reducing dimensions of the search space



PG: 2D Search

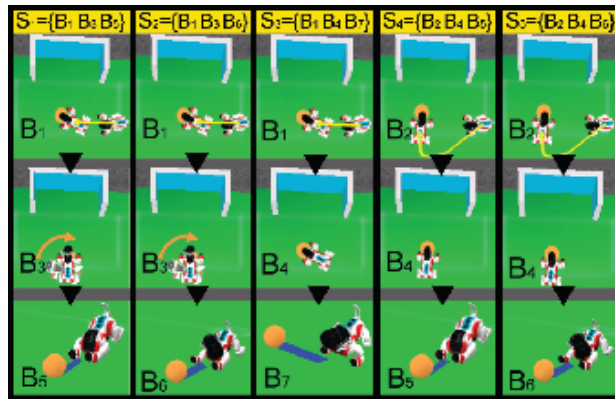
EPG: 1D Search

Implementation

B_1, B_2 : *ball approaching* (11 parameters)

B_3, B_4 : *ball controlling* (4 parameters)

B_5, B_6, B_7 : *kick actions* (23 parameters)



- Objective function

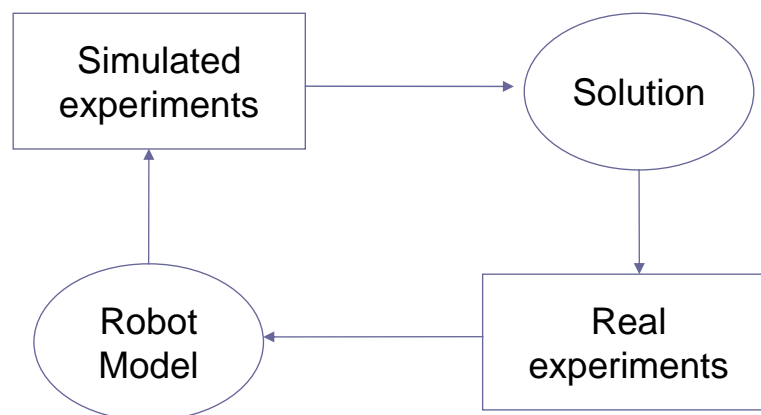
linear combination of

- quality of walking gait
- quality of approaching ball
- quality of kick

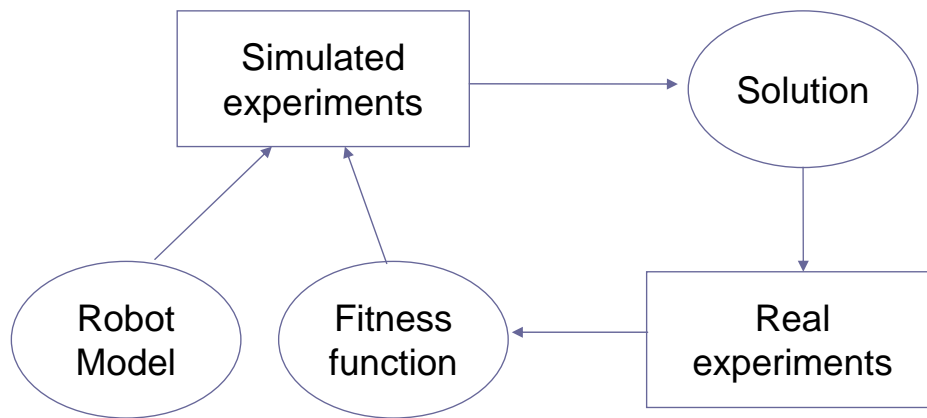
- Comparison with PG

[Kohl and Stone 2004]

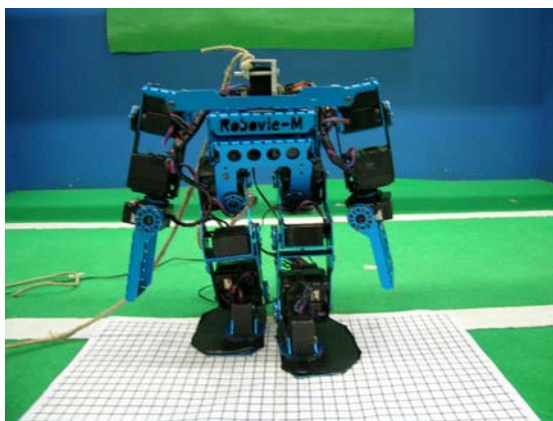
Second case study: Learning actions interleaving simulated and real data



Proposed approach



Implementation on real and simulated robots



Real robot



Simulated robot in USARSim

Walking and **Kicking** implemented with a fuzzy controller based on oscillation of torso and legs and arms swing [*Thanks to Univ. of Padova*]

Humanoid Walking



Problems

- Difficult task (20 DOF)
- Noise experiments
- Difficulty in operating the robot

Solution and Results

(next talk)

How to learn with OpenRDK

Typical learning process

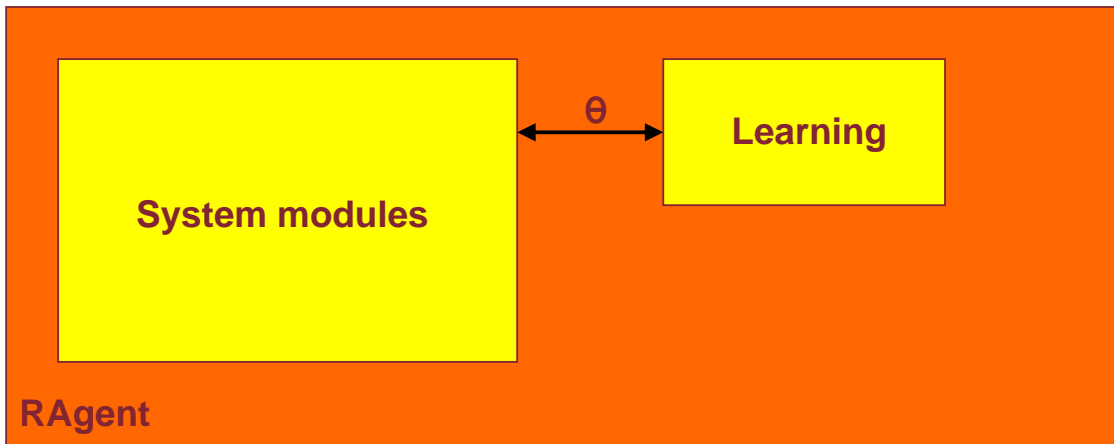
- Try and evaluate process
- Execute many runs with different parameters and measure the performance
- Exploiting a simulator

Advantages of using OpenRDK

- implement learning on top of existing modules
- learning parameters exported as repository properties
- Modularity w.r.t. learning methods and robot model (real or simulated)
- no need to recompile the modules!

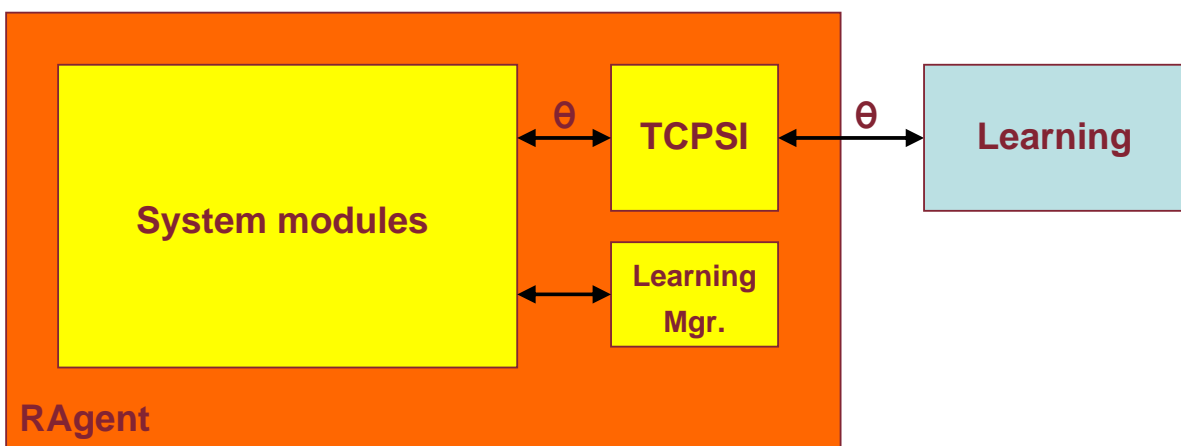
How to learn with OpenRDK

Learning in an OpenRDK module



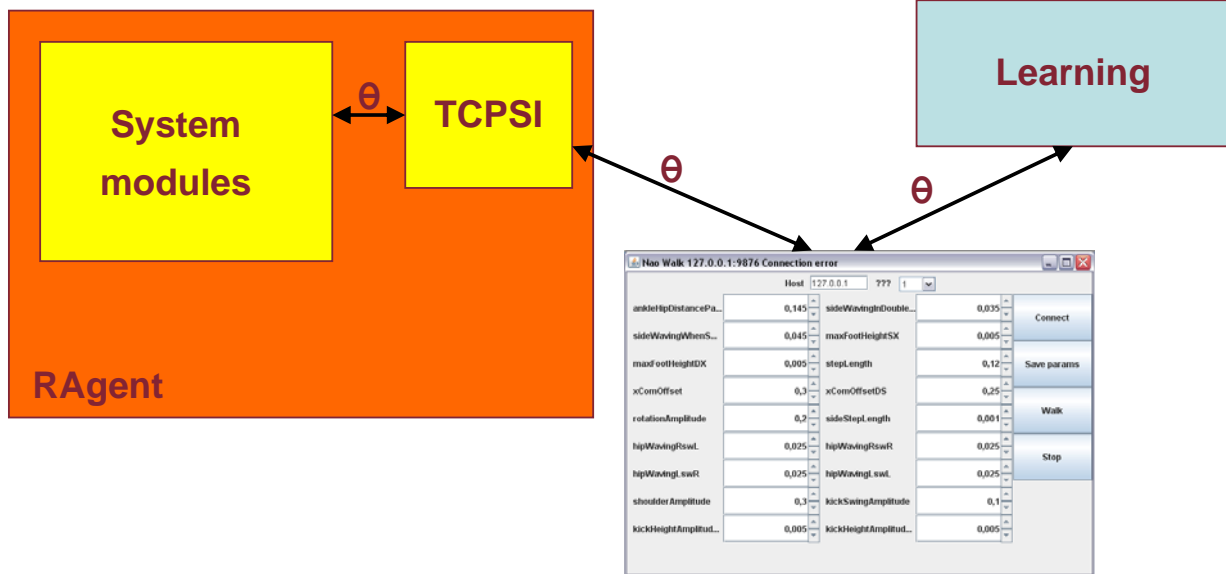
How to learn with OpenRDK

Learning in an external application



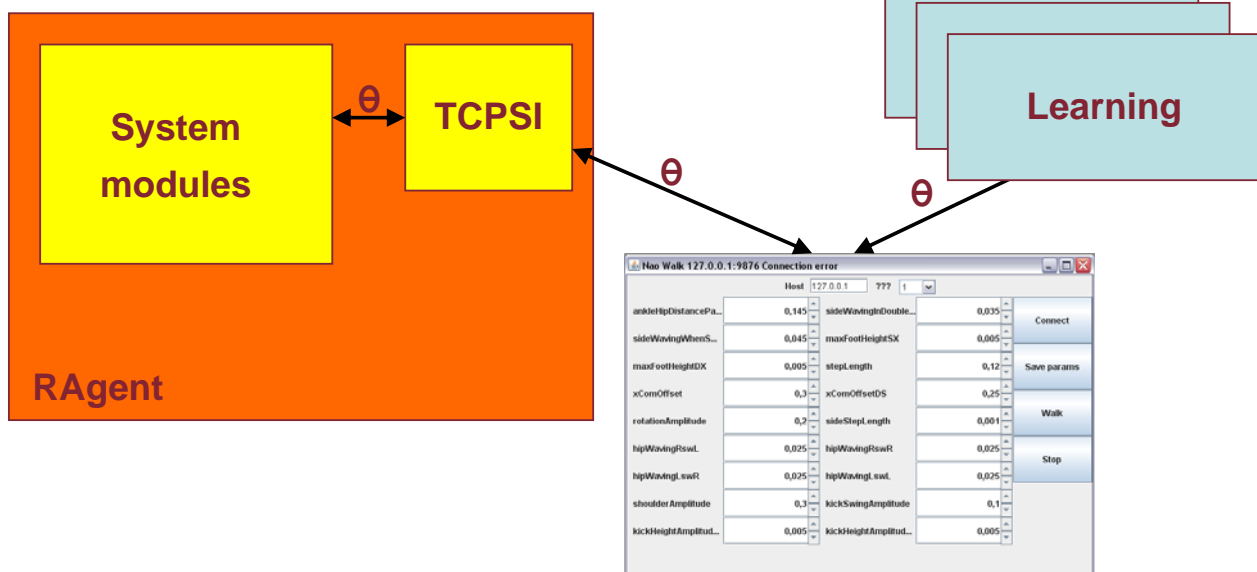
How to learn with OpenRDK

Controlling the learning process



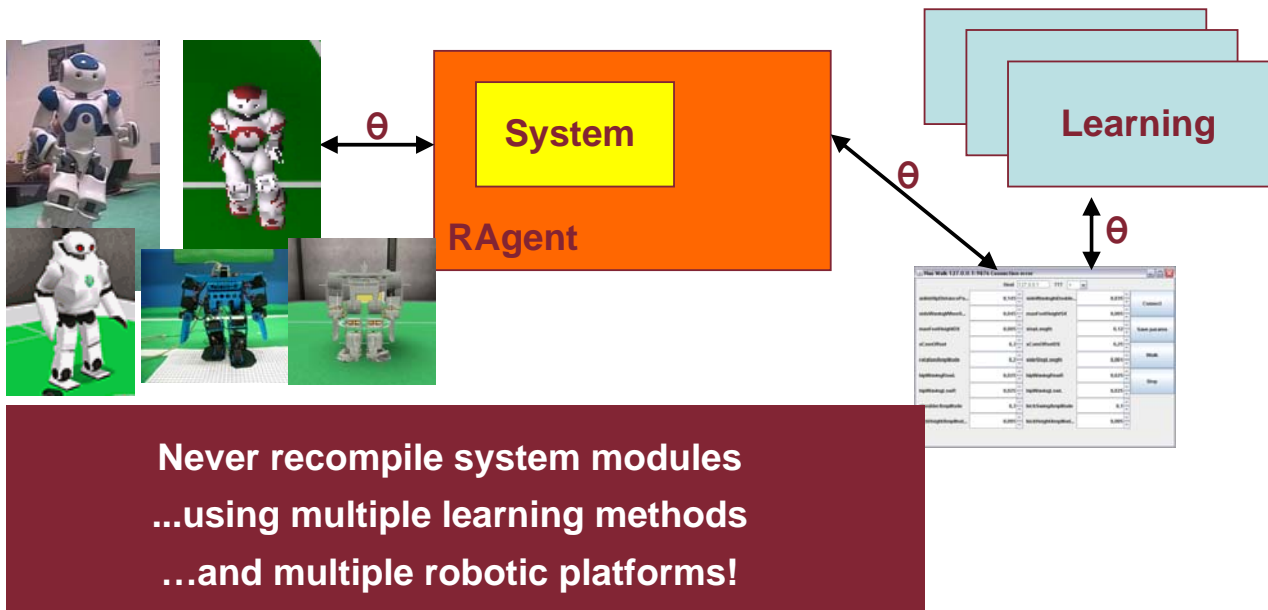
How to learn with OpenRDK

Modularity w.r.t. learning methods



How to learn with OpenRDK

Modularity w.r.t. (real and simulated) robot models



References

A. Cherubini, F. Giannone, L. Iocchi, M. Lombardo, and G. Oriolo.

Policy Gradient Learning for a Humanoid Soccer Robot.

In Robotics and Autonomous System - Special Issue on "Humanoid Soccer Robots". 2009.

A. Cherubini, F. Giannone, L. Iocchi, and P. F. Palamara.

An extended policy gradient algorithm for robot task learning.

In Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2007.

A. Cherubini, F. Giannone, and L. Iocchi.

Layered Learning for a soccer legged robot helped with a 3D simulator.

In RoboCup 2007: Robot Soccer World Cup XI. 2008. pp. 385-392.

L. Iocchi, F. Dalla Libera, and E. Menegatti.

Learning Biped Locomotion using Interleaved Simulated and Real Data.

In Proc. of 2nd Workshop on Humanoid Soccer Robots, 2007.