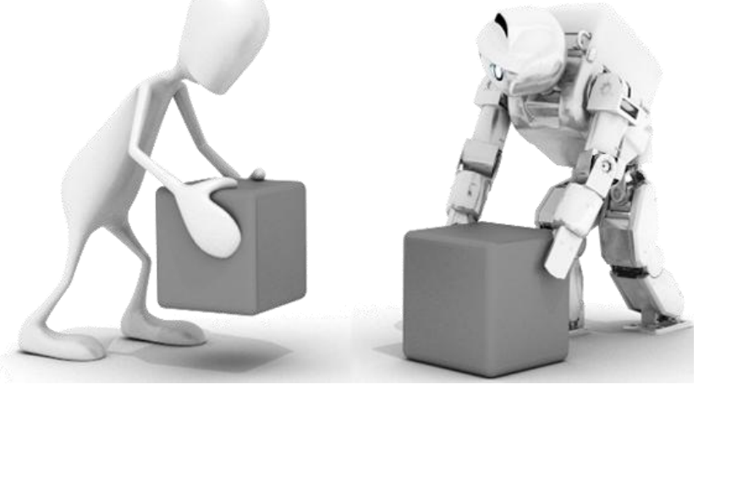
MULTI-OPTIMA EXPLORATION WITH ADAPTIVE GAUSSIAN MIXTURE MODEL



Sylvain Calinon[1], Affan Pervez[2] and Darwin G. Caldwell[1]

[1] Learning & Interaction Lab, Department of Advanced Robotics (ADVR), Italian Institute of Technology (IIT)

[2] KTH Royal Institute of Technology

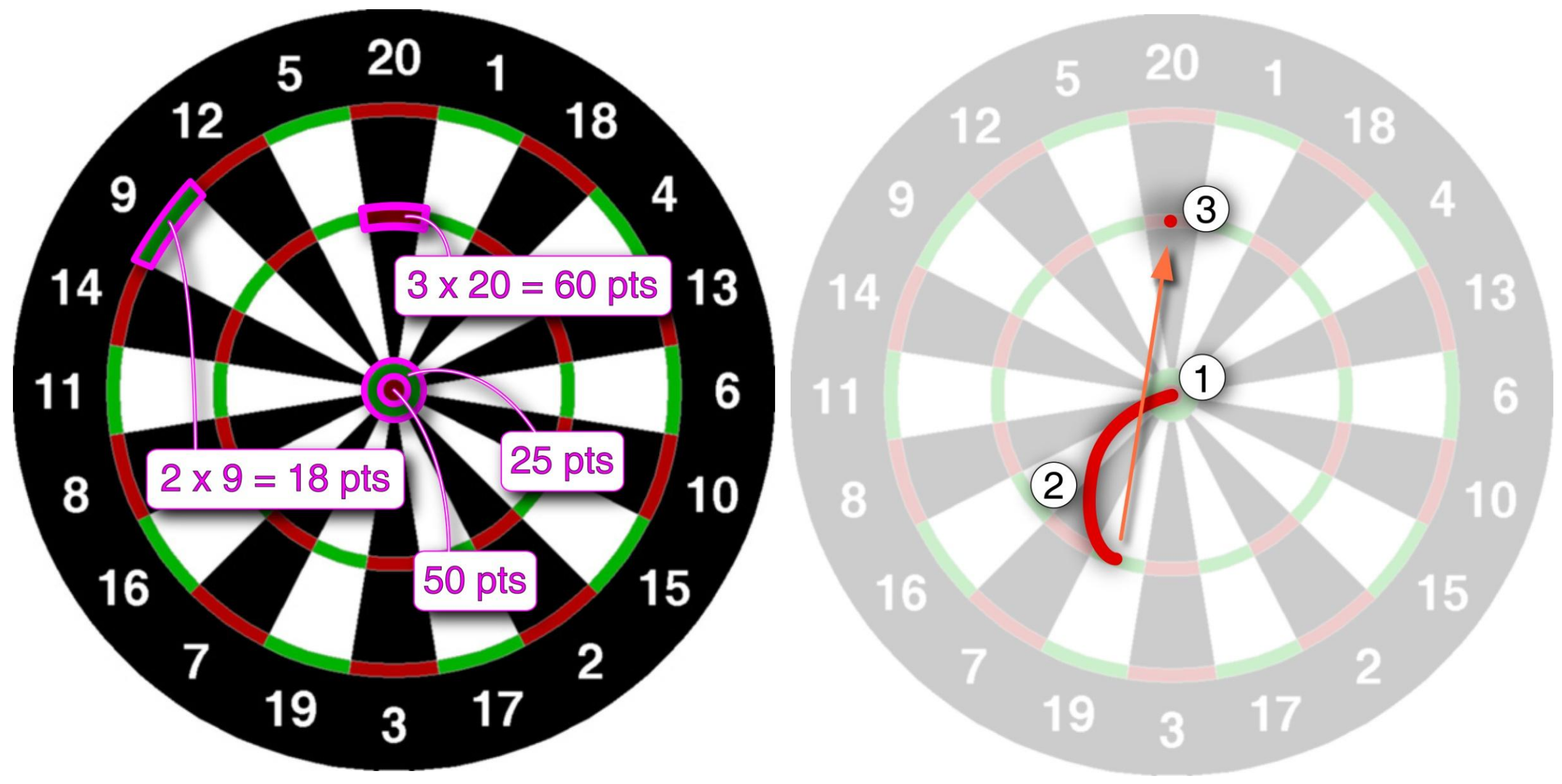
Abstract

In learning by exploration problems such as reinforcement learning (RL), direct policy search, stochastic optimization or evolutionary computation, the goal of an agent is to maximize some form of reward function. Often, these algorithms are **designed to find a single policy solution**. We address the problem of representing the space of control policy solutions by **considering exploration as a density estimation problem.**

Such representation provides additional information such as **shape and curvature of local peaks** that can be exploited to analyze the discovered solutions and guide the exploration. We show that the search process can easily be **generalized to multi-peaked distributions** by employing a **Gaussian mixture model (GMM)** with an adaptive number of components. The GMM has a dual role: representing the space of possible control policies, and guiding the exploration of new policies.

The approach is tested in a dart game experiment formulated as a black-box optimization problem, where the agent's throwing capability increases while it chases for the best strategy to play the game. This experiment is used to study how the proposed approach can **exploit promising solution alternatives** in the search process, where the optimality criterion slowly drifts over time. The results show that the proposed multi-optima search approach can anticipate such changes by exploiting promising candidates to **smoothly adapt to the change of global optimum.**

Which point on the dartboard should a player aim at in order to maximize his/her scores?



Repartition of scores on a standard dartboard

Path of the global optimum for a progressively increasing throwing accuracy

The triple-20 is the maximum reachable score on the dartboard, but in the long run, the position to target depends on the throwing skills of the player:

Player 1 (beginner) should aim at the bullseye.
Player 2 (intermediate) should aim at a point in the lower-left part.
Player 3 (expert) should aim at the triple 20.

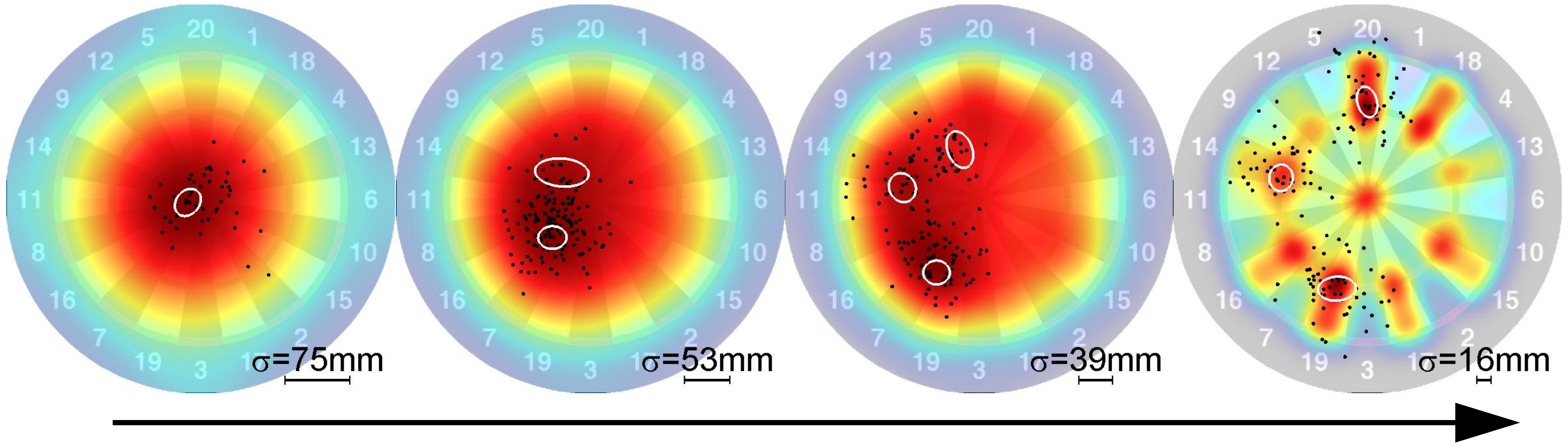
1

2

3

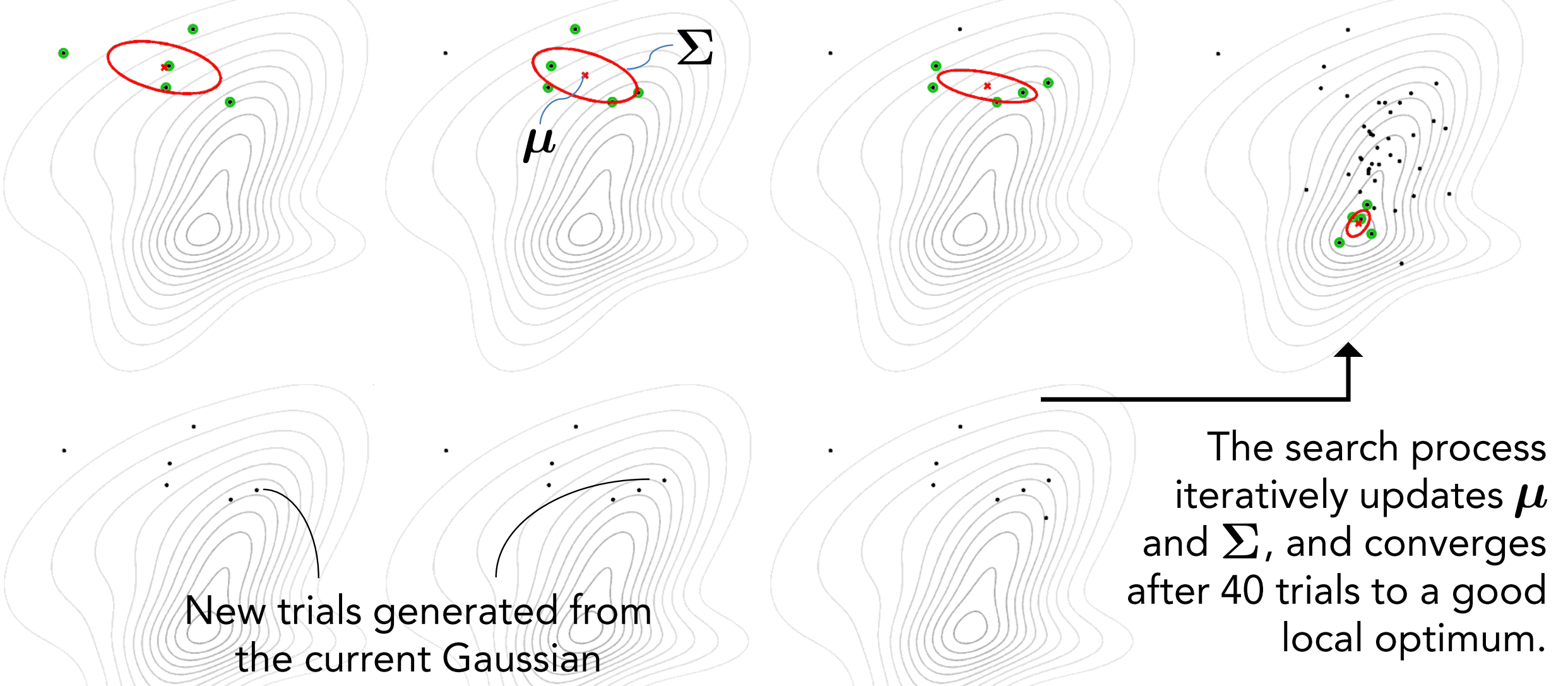
Motivations

- When growing, gaining experience, recovering from an injury or practicing a sport, our motor capabilities continuously change.
- Similarly, the capabilities or objectives of a robot agent can change over time, requiring learning strategies that can continuously adapt to these fluctuations without requiring the user to explicitly trigger exploration/exploitation behaviors.
- A parallel exploration of multiple policy options might provide the robot agent with a robust way to adapt to changing environments, changing body morphology or changing perception-action capabilities through its developmental lifespan.



When the player's throwing skill accuracy improves, the solution space (colored heatmap, unknown to the agent) slowly drifts from a single peak to a multimodal distribution. The proposed learning algorithm aims at keeping track of multiple options (white Gaussians) to swiftly adapt to the development of the agent during the search process.

Reward-weighted learning approaches



New trials generated from the current Gaussian

The search process iteratively updates $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and converges after 40 trials to a good local optimum.

- Several search procedures in stochastic optimization, evolutionary computation or reinforcement learning (RL) rely on an expectation-maximization (EM) process to iteratively update a policy together with the exploration noise used to generate new trials.
- Examples of such approaches are the cross-entropy method (CEM), the covariance matrix adaptation evolution strategy (CMA-ES), and RL approaches exploring directly in the policy parameters space such as PoWER (policy learning by weighting exploration with the returns) and PI^2 (policy improvement with path integrals).

[D.P. Kroese *and* R.Y. Rubinstein (2004) "The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning", Springer]

[N. Hansen (2006), "The CMA evolution strategy: a comparing review", Towards a new evolutionary computation, Springer]

[J. Kober *and* J. Peters (2010) "Imitation and reinforcement learning: practical algorithms for motor primitives in robotics", IEEE Robotics and Automation Magazine 17:2]

[F. Stulp *and* O. Sigaud (2012) "Path integral policy improvement with covariance matrix adaptation", Intl Conf. on Machine Learning (ICML)]

- Covariance information can serve several purposes. First, it can guide the exploration by defining an adaptive exploration-exploitation trade-off. Then, it conveys important information about the neighborhood of the policy solutions (e.g., shape, total surface, principal directions, curvature).
- In some tasks, the immediate neighborhood of the solution manifold has different curvature for different local optima, making some regions more tolerant to errors than others. The solution manifold can for example be characterized by a continuous portion of space in which the reward is maximum. There is thus no global optimum based on the reward information alone, but it is the local spread of the region that determines the best solution that one can reach.
- Most search algorithms are designed to locate a single optimal point, which does not seem to match with the way humans learn skills.

[D. Sternad, M.O. Abe, X. Hu *and* H. Mueller (2011) "Neuromotor noise, error tolerance and velocity-dependent costs in skilled performance", PLoS Computational Biology 7:9]

Extension of the search process to a mixture of Gaussians

- The process can be extended to the search of local solutions with complex and asymmetric shapes that could not be approximated efficiently by a single Gaussian, and to multi-optima subspaces to let the agent select alternative options with respect to the current situation (e.g., based on space restriction, occlusion, injured articulation or fatigued muscles).
- It also allows the agent to adapt to progressively changing environment (slowly drifting reward functions). In this case, the system can keep track of regions that might at a given time have slightly lower reward, but that can potentially lead to optimal solutions in the future.
- In contrast to EM in standard GMM estimation, the weighting mechanism does not only consider the probability of belonging to one of the mixture component, but also the rewards of the different trials:

E-step:

$$h_i(\boldsymbol{\Theta}_m) = \frac{\pi_i\,\mathcal{N}\Big(\boldsymbol{\Theta}_m|\,\boldsymbol{\mu}_i^{(n-1)},\boldsymbol{\Sigma}_i^{(n-1)}\Big)}{\sum_k^K \pi_k\,\mathcal{N}\Big(\boldsymbol{\Theta}_m|\,\boldsymbol{\mu}_k^{(n-1)},\boldsymbol{\Sigma}_k^{(n-1)}\Big)}.$$

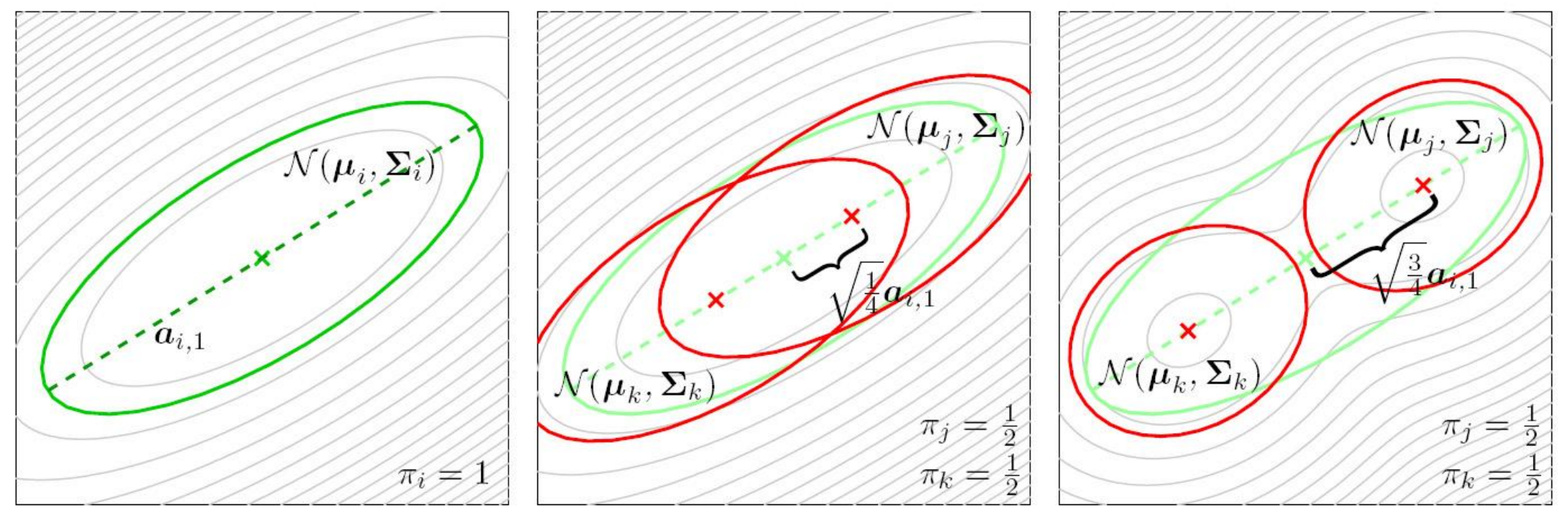
M-step:

$$\boldsymbol{\mu}_i^{(n)} = \boldsymbol{\mu}_i^{(n-1)} + \frac{\sum_m^M r(\boldsymbol{\Theta}_m)h_i(\boldsymbol{\Theta}_m)\,\Big[\boldsymbol{\Theta}_m - \boldsymbol{\mu}_i^{(n-1)}\Big]}{\sum_m^M r(\boldsymbol{\Theta}_m)h_i(\boldsymbol{\Theta}_m)},$$

$$\boldsymbol{\Sigma}_i^{(n)} = \frac{\sum_m^M r(\boldsymbol{\Theta}_m)h_i(\boldsymbol{\Theta}_m)\Big[\boldsymbol{\Theta}_m - \boldsymbol{\mu}_i^{(n-1)}\Big]\Big[\boldsymbol{\Theta}_m - \boldsymbol{\mu}_i^{(n-1)}\Big]^\top}{\sum_m^M r(\boldsymbol{\Theta}_m)h_i(\boldsymbol{\Theta}_m)} + \boldsymbol{\Sigma}_0,$$

$$\pi_i^{(n)} = \frac{\sum_m^M r(\boldsymbol{\Theta}_m)h_i(\boldsymbol{\Theta}_m)}{\sum_k^K\sum_m^M r(\boldsymbol{\Theta}_m)h_k(\boldsymbol{\Theta}_m)}.$$

$\{\boldsymbol{\Theta}_m\}_{m=1}^M$ is the ordered set of the best policy parameters for each component i and for the last L trials, with $r(\boldsymbol{\Theta}_1)h_i(\boldsymbol{\Theta}_1)\geq r(\boldsymbol{\Theta}_2)h_i(\boldsymbol{\Theta}_2)\geq\ldots\geq r(\boldsymbol{\Theta}_M)h_i(\boldsymbol{\Theta}_M)$. $\boldsymbol{\Sigma}_0$ is a minimum exploration noise avoiding premature convergence to poor local optima.



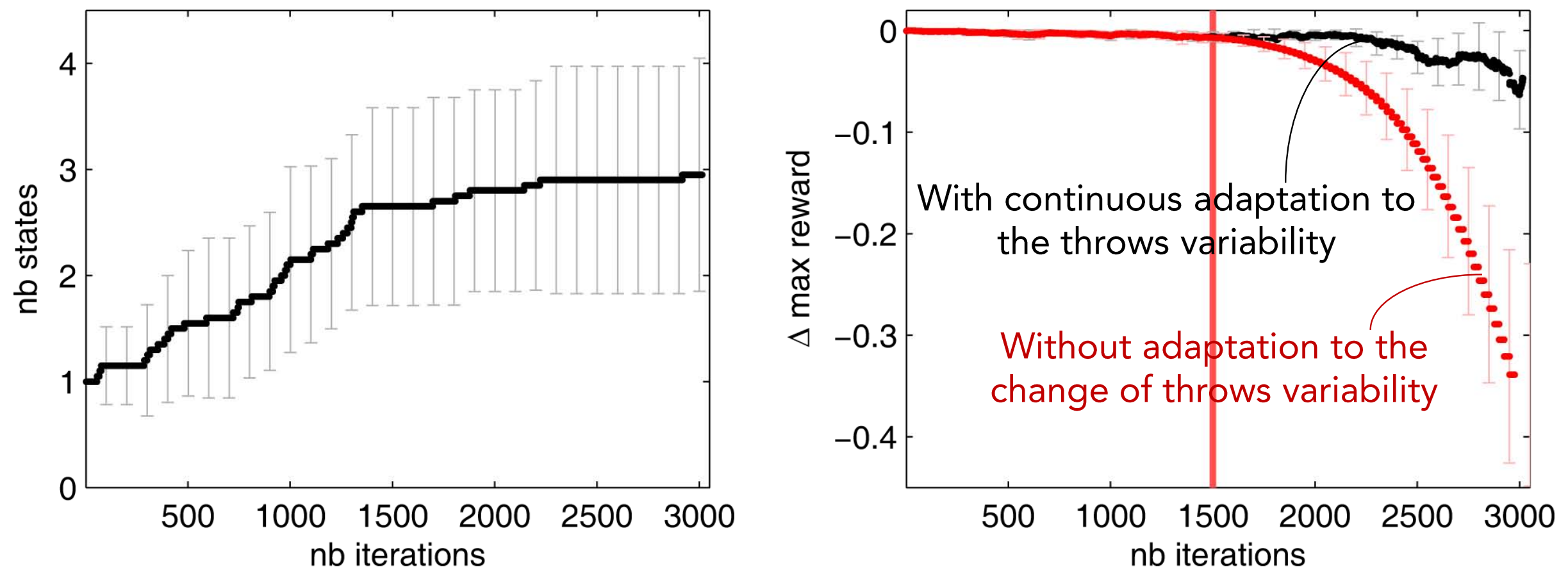
$\mathcal{N}(\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$
$\boldsymbol{a}_{i,1}$
$\pi_i = 1$
$\mathcal{N}(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$
$\mathcal{N}(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$
$\sqrt{\frac{1}{4}}\boldsymbol{a}_{i,1}$
$\pi_j = \frac{1}{2}$
$\pi_k = \frac{1}{2}$
$\mathcal{N}(\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$
$\sqrt{\frac{3}{4}}\boldsymbol{a}_{i,1}$
$\mathcal{N}(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$
$\pi_j = \frac{1}{2}$
$\pi_k = \frac{1}{2}$

During the search process, Gaussians are split along their principal axis if the increase of components has a significant impact on the likelihood. Two Gaussians are merged if the removal of one component only slightly decreases the likelihood.

[Z. Zhang, C.Chen, J. Sun *and* K.L. Chan (2003) "EM algorithms for Gaussian mixtures with split-and-merge operation", Pattern Recognition 36:9]

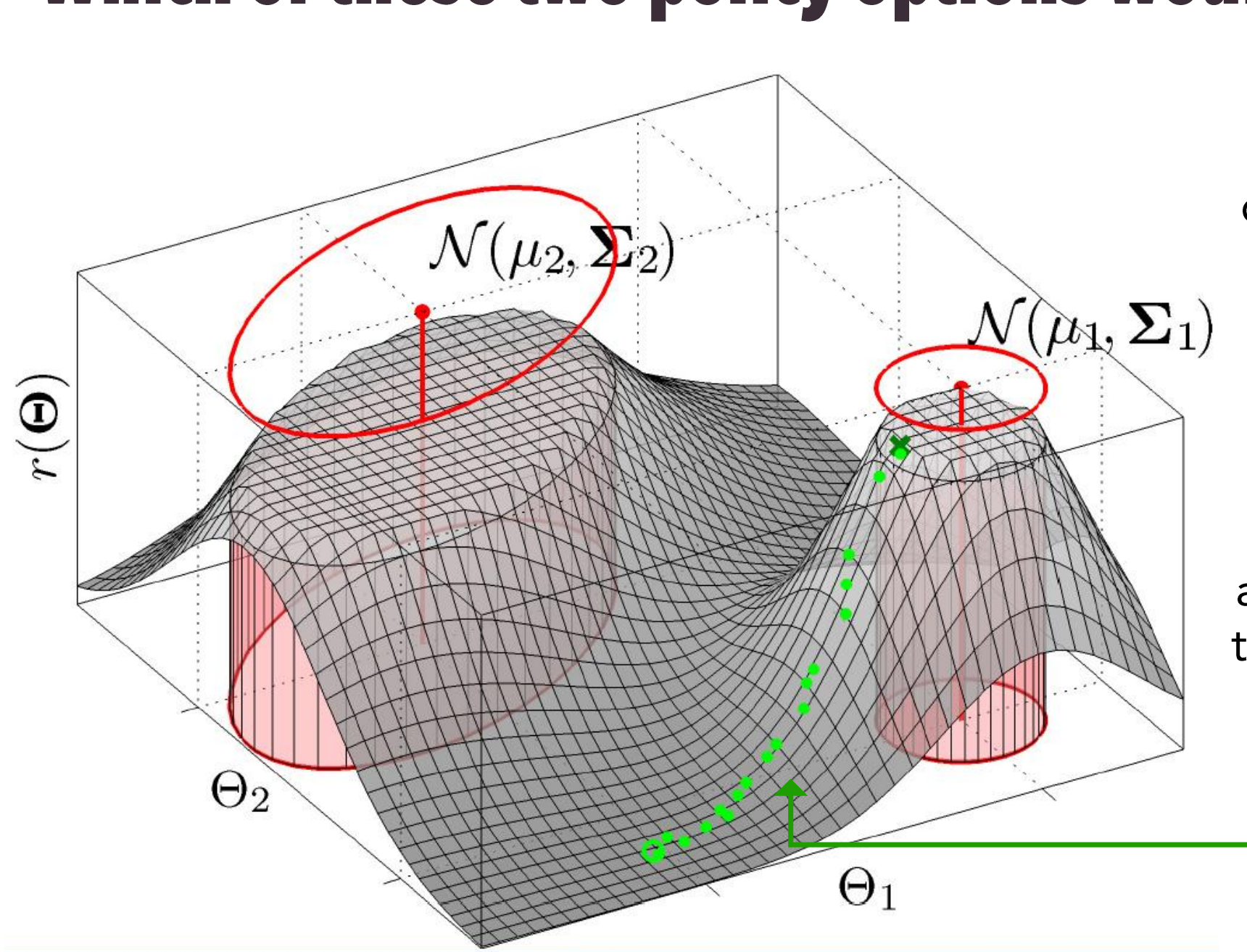
Experimental results

- The agent refines its throwing skill while searching for the regions of the dartboard where it should aim to obtain high scores.
- The progressive reshaping of the solution space requires the agent to balance exploration and exploitation in a continuous manner, incorporating a developmental stance to the exploration behavior.
- Even though the complexity of the solution space increases (from single to multi-peaked distribution), the agent is capable of maintaining a decent score (black curve below has small linear decay trend) by creating policy alternatives. These alternatives can cope with the discontinuous switches of global optima when the throws get more accurate.
- In contrast, if the agent stops adapting its strategy while still improving its throwing capability, its performance in the game quickly degrades (see red curve below with exponential decay trend).



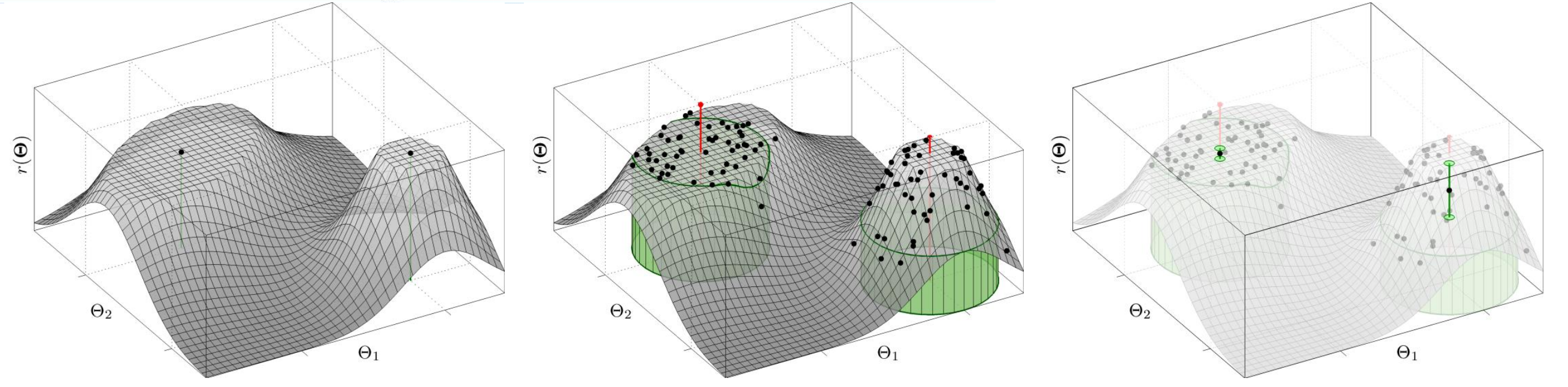
Left: Average number of Gaussians in the GMM (results averaged over 20 runs). **Right:** Maximum reward computed by using the centers of the Gaussians with $\hat{r}=\max_i r(\boldsymbol{\mu}_i)$. The obtained rewards are compared to the theoretical global optimum computed for each throws accuracy. The black curve shows the difference between this best theoretical reward $r^{\max}$ and the reward $\hat{r}$ computed from the discovered policy option(s) ($\Delta r=\hat{r}-r^{\max}$). The red line shows the effect of stopping the adaptation to the throwing skills improvement after 1500 iterations.

Which of these two policy options would you choose?



$\mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2)$
$\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1)$
$r(\boldsymbol{\Theta})$
Θ_2
Θ_1

A standard gradient-ascent process starting from the green circle would converge to one of the two peaks (green cross). We are interested in search procedures that approximate regions of high rewards with a density function, such that the algorithm does not only detect a peak and stops, but continues the exploration to provide more information about the viability, spread and curvature of the solution subspaces.



If the agent is very precise, the policy parameters $\boldsymbol{\Theta}=\boldsymbol{\mu}_1$ provide a higher reward. But in a real situation, the precision can vary depending on the agent, the environment and the context. For a given level of noise $\hat{\boldsymbol{\Sigma}}$, the agent should select $\boldsymbol{\Theta}=\boldsymbol{\mu}_2$ to maximize the average reward. $\boldsymbol{\Theta}=\boldsymbol{\mu}_1$ is more risky but will some times lead to higher rewards.

[S. Calinon, P. Kormushev *and* D.G. Caldwell (2012) "Compliant skills acquisition and multi-optima policy search with EM-based reinforcement learning", Robotics and Autonomous Systems]

SEVENTH FRAMEWORK PROGRAMME

Pandora
Persistently Autonomous Robots

STIFF flop

Contact e-mail: sylvain.calinon@iit.it

Programming-by-demonstration.org