Multi-optima exploration with adaptive Gaussian mixture model

Sylvain Calinon, Affan Pervez and Darwin G. Caldwell

***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 (or minimize a cost 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. A variation of expectation-maximization (EM) applied to reward-weighted policy parameters is presented to model the space of possible solutions, as if this space was a probability distribution. 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 new promising solution alternatives in the search process, when 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.**

I. INTRODUCTION

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. This form of continuous self-calibration is particularly important for systems designed to work in unpredictable environments where the agent needs to adapt an existing skill to new situations.

Most search algorithms are designed to locate a single optimum, which does not seem to match with the way humans learn skills [1]. For example, elite sport athletes can practice and improve their running performance by relying on different policies, favoring two different categories of local optima corresponding to either long strides or high cadence [2]. Similarly, a parallel exploration of multiple policy options might provide a robotic agent with a robust way to adapt to changing environments, changing body morphology or changing perception-action capabilities through its developmental lifespan.

We propose to study the multi-optima search issue in a simulated dart game experiment formulated a black-box optimization problem (the policy consists here of a single state with an action parameterized by two variables). The agent progressively increases its throwing skills while looking for appropriate strategies to win the game. A variety of skillful games characterized by throws of objects at a target have been studied, by providing a simple way to formulate the reward function as an explicit score. Most of the games settings studied so far were characterized by given position(s) to reach on the target [3]–[5]. Lawrence *et al* [3] studied darts throwing by aiming at the bullseye. Kober *et al* [4] explored the problem of throwing darts at pre-specified positions on the dartboard (*Around the clock* game rule). Da Silva *et al* [5] studied the problem of learning parameterized skills by reaching for the center of a target placed at different locations.

In this paper, we consider the problem of discovering how to obtain the maximum score on an official dartboard (Fig. 1), split into 20 sectors with scores arranged in a given unordered sequence. The sectors are radially split with two bands corresponding to areas doubling (outer band) and tripling (inner band) the score of the sector. Two additional rings at the center define regions of 25 and 50 points. This arrangement is more interesting than radially increasing scores because the best position that a player should target takes various forms. One might think at first sight that aiming at the area of maximum score on the board (the triple-20) is the optimal strategy. In reality, this position varies with the precision at which the player can throw the dart. Beginners should aim at the bullseye. When improving their throwing skills, this point is progressively displaced with a nonlinear path to a region in the bottom-left part of the dartboard. When the players become very skilled, this point "jumps" to the triple-20 score region.

This provides an interesting framework to study developmental learning perspectives, where the highly rewarded regions vary from simple single peaked distributions to a variety of multi-peaked distributions, see Fig. 1. For a given throwing accuracy, the distribution of expected scores on the dartboard can be analytically evaluated [6]. This provides us with ground truth data to evaluate how the artificial agent refines its policy by preserving alternative options.

The solution space of this problem is characterized by a single global optimum at different locations on the dartboard,

S. Calinon and D.G. Caldwell are with the Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. {sylvain.calinon, darwin.caldwell}@iit.it.

A. Pervez is with the KTH Royal Institute of Technology, Stockholm, Sweden. affan@kth.se.

This work was partially supported by the PANDORA European project (FP7-ICT-288273) and the STIFF-FLOP European project (FP7-ICT-287728).

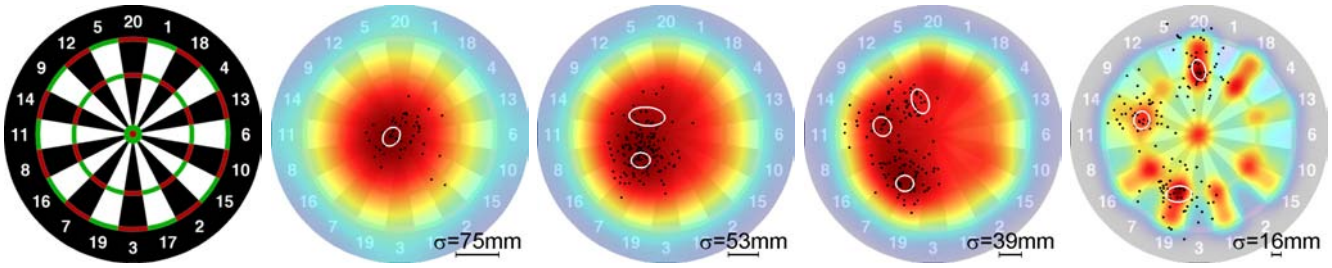

Fig. 1. Evolution of the search process. The colored heatmaps (corresponding to steps 1, 1060, 1757 and 3000) represent the expected distribution for the current throws accuracy of the agent, with colors from cyan to dark red linearly spread between lowest and highest scores. Steps 1060 and 1757 are the steps when Gaussian splits occurred. The black dots show the last $L = 150$ trials. The white ellipses represent the mixture of Gaussians representing the promising options discovered by the agent (approximation of the unknown solution landscape by iterative exploration and adaptation).

where the multi-optima exploration process can be exploited to keep track of promising alternative candidates to robustly adapt to throwing skills improvement.

II. PROPOSED APPROACH

The search problem is formulated as an incremental approximation of the space of control policy solutions, represented in the form of a probability distribution. The rewards act as likelihoods of the policy parameters, which can be achieved by defining positive reward functions (e.g., formulated in exponential form). The distribution is approximated by a *Gaussian mixture model* (GMM) with an adaptive number of Gaussians. The model has a dual role: representing the space of possible control policy, and guiding the exploration of new policies.

In order to estimate such distribution, we present an *expectation-maximization* (EM) algorithm applied to weighted datapoints, and a *split&merge* procedure to adapt the number of components in the model. It is used to fit the best samples obtained so far (weighted by their rewards) and generate new samples in an incremental manner.

A. EM-based search procedures

A tremendous effort within the machine learning and robotics community has been directed to moving *reinforcement learning* (RL) to continuous real-world domains. A promising way is to define parameterized policies and explore directly in the policy parameters space, by transforming the gradient estimation procedure into a probabilistic estimation problem [7]–[10]. The principal idea is to treat positive rewards as probabilistic weights, where an EM procedure can then be used to iteratively optimize the policy parameters.

Dayan and Hinton originally suggested that a RL problem can be tackled by EM to avoid gradient computation [11]. They introduced the core idea of treating immediate rewards as probabilities of a fictitious event, in which case probabilistic inference techniques can be used for optimization. From this simple idea, various reward-weighted policy learning approaches emerged. The reason of this success can be attributed to the rapid convergence rate that is well suited to real-world operational requirements, and to the compatibility with other research perspectives aimed at designing compact and robust representation for parameterized policy [5], [7], [12]–[16].

In *policy learning by weighting exploration with the returns* (PoWER) [10], the action is treated as an unobserved variable and the returns are considered as a probability distribution. It relies on the following insight: a safe way to generate new policies is to look in the convex combination of sampled policies. PoWER estimates a policy $\boldsymbol{\Theta}^{(n)}$ at iteration n, such as to maximize the lower bound on the expected return from following the policy. In its simplest form, and in the case of episodic rewards (single final rewards), a new policy can be estimated and added to the training set by following the update

$$\boldsymbol{\Theta}^{(n)} = \boldsymbol{\Theta}^{(n-1)} + \frac{\sum_{m}^{M} r(\boldsymbol{\Theta}_m)\left[\boldsymbol{\Theta}_m - \boldsymbol{\Theta}^{(n-1)}\right]}{\sum_{m}^{M} r(\boldsymbol{\Theta}_m)}, \quad (1)$$

where an ordered set of the best policies $\{\boldsymbol{\Theta}_k\}_{k=1}^{M}$ obtained so far with $r(\boldsymbol{\Theta}_1) \geq r(\boldsymbol{\Theta}_2) \geq \ldots \geq r(\boldsymbol{\Theta}_M)$ is used at each iteration n as a form of importance sampling [10].

The above equation is written in a form that emphasizes correspondences with gradient-based approach. $\left[\boldsymbol{\Theta}_m - \boldsymbol{\Theta}^{(n-1)}\right]$ represents the relative exploration between the policy parameters used in the m-th ordered trial and the current best policy parameters. This difference is weighted by the corresponding reward $r(\boldsymbol{\Theta}_m)$, and normalized by the sum of the other rewards. The equation above can also be rewritten as a weighted sum of the best policies obtained so far, thus making links with imitation learning strategies where each iteration in the process corresponds to the imitation of the most successful policy parameters. In the case of episodic rewards, it also has direct links with state-of-the-art algorithms in stochastic optimization and evolutionary computation such as the *cross-entropy method* (CEM) [17] and the *covariance matrix adaptation evolution strategy* (CMA-ES) [18], see also [14].

B. Exploiting covariance information in the exploration problem

When searching for new solutions, large exploration noise can lead to faster convergence due to greater changes of the mean policy. Low exploration noise can in contrast converge to more accurate solutions, and avoid bringing the robot into

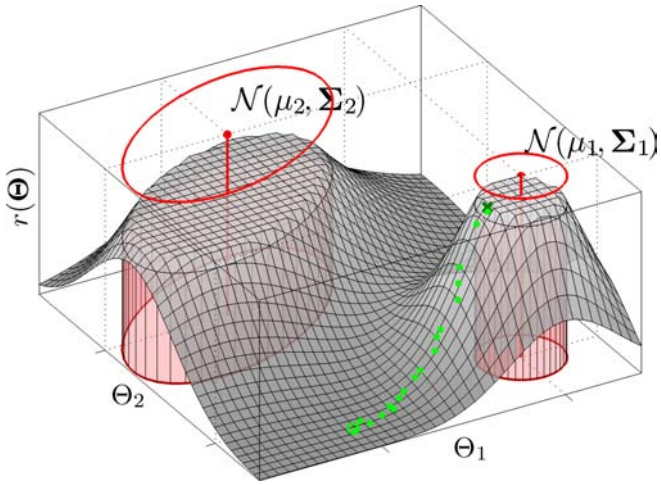

Fig. 2. Illustration of an exploration problem characterized by several local optima. The solution landscape is represented by two policy parameters $\boldsymbol{\Theta} = [\Theta_1, \Theta_2]^\top$ space, with the elevation proportional to the rewards $r(\boldsymbol{\Theta})$. In this illustrative example, two local optima could be found by the robot, characterized by centers $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2\}$ and covariance matrices $\{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2\}$. A standard gradient-ascent procedure starting from the green circle could converge to one of the two peaks and possibly stop after attaining a local maximum (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 and spread of the solution subspaces. In this example, the reward $r(\boldsymbol{\mu}_1)$ is slightly higher than $r(\boldsymbol{\mu}_2)$. However, $\boldsymbol{\mu}_2$ can be considered as a safer choice for the policy because the larger covariance $\boldsymbol{\Sigma}_2$ results in higher tolerance to errors.

unsafe regimes. The exploration thus has to be sufficiently rich to converge in a reasonable amount of time and avoid getting stuck in poor local optima. A common practice is to introduce decaying exploration terms, where the noise is often generated independently for each variable (without covariation). Alternatively, the EM process can be used to optimize exploration noise together with policy parameters [7], similarly as in stochastic optimization and evolutionary search problems [17], [18].

The exploration noise can be expressed in the form of a covariance matrix and updated with

$$\boldsymbol{\Sigma}^{(n)} = \frac{\sum_{m}^{M} r(\boldsymbol{\Theta}_m) \left[\boldsymbol{\Theta}_m - \boldsymbol{\Theta}^{(n-1)}\right]\left[\boldsymbol{\Theta}_m - \boldsymbol{\Theta}^{(n-1)}\right]^\top}{\sum_{m}^{M} r(\boldsymbol{\Theta}_m)} + \boldsymbol{\Sigma}_0, \tag{2}$$

where $\boldsymbol{\Sigma}_0$ is a regularization term (diagonal covariance matrix) corresponding to a minimum exploration noise used to avoid premature convergence to poor local optima.

Equations (1) and (2) form at each iteration a multivariate normal distribution. The 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), see Fig. 2.

In the case of human movements, it was shown that redundancy provides a way to cope with noise at a motor level by channeling the noise in directions that have minimal effect on achieving the task goal [19]. Skilled performers may take advantage of this redundancy and align their actions with the solution manifold corresponding to a given task goal, i.e., the space in which noise and variability have little or no effect on the end result. Sternad *et al* [19] suggested that variability may offer a way to quantify error tolerance via the shape of the result function. Indeed, one reason to prefer some locations over others in the task solution manifold is the sensitivity of the result to variability (tolerance to errors). 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 we would like to reach, see Fig. 2.

The formulation of the search problem as a global approximation of the space of control policy solutions introduces new possible considerations for the selection of policy parameters. The selection does not only need to be driven by the reward value but can also include other (possibly context-dependent) factors such as tolerances to errors. The EM-based RL framework appears to be a good candidate to determine and evaluate the robustness of such solutions by analyzing how the policy parameters can exploit the redundancy and sensitivity to errors. It allows the study of policy improvement strategies that do not only consist of reaching a locally optimum solution, but that are capable of approximating locally the solution space. This can be used to determine the possible range of noise that can be introduced without perturbing the outcome of the task, and to analyze the correlations among the policy parameters for the reproduction.

In the next section, we show that the problem can be extended to a mixture of policy subspaces. Namely, to the search of a solution space by fitting a density function that does not (necessary) have a single optimum. More complex solution space can thus be considered with multimodal distributions learned in an adaptive and incremental manner.

The representation of density functions as mixture models can be applied in various manners within an exploration context. Botev and Kroese presented a *generalized cross-entropy* (GCE) method applied to the problem of density estimation, by making links between kernel-based and mixture-based approaches [20]. They tackled density estimation as a *functional* optimization problem (in contrast to solving a *parametric* optimization problem), and showed that by selecting an appropriate divergence measure, the approach is equivalent to choosing a discrete mixture of kernel functions as sampling density. Kobilarov explored the use of GMM for stochastic sampling in CEM [21], and showed that, for exploration in a parameterized trajectory space, the use of GMM can improve the performance of sampling-based motion planning with a more global exploration of feasible

solutions.

C. Multi-optima exploration

The update equations in (1) and (2) are now extended to the autonomous search of multiple local optima with an EM procedure. Intuitively, the process can be visualized as fitting a *Gaussian mixture model* (GMM) in which a set of normal distributions characterized by centers $\boldsymbol{\mu}_i$, covariance matrices $\boldsymbol{\Sigma}_i$ and priors (mixing coefficients) π_i are iteratively used for refinement and generation of new trials. The process corresponds to a reward-weighted stochastic optimization process, where each $\boldsymbol{\mu}_i$ represents the best local guess of the policy parameters, while $\boldsymbol{\Sigma}_i$ represents the spread of the region in this policy parameters space. A prior (mixing coefficient) π_i is associated with each Gaussian to encode the importance of each local solution subspace.

In contrast to the use of EM in a standard GMM fitting mechanism, the weighting process does not only consider the probability of belonging to one of the mixture component, but also the rewards of the different samples. Namely, at iteration n, we have for each component $i \in \{1, \ldots, K\}$

E-step:

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

M-step:

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

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

$$\pi_i^{(n)} = \frac{\sum_m^M r(\boldsymbol{\Theta}_m) h_i(\boldsymbol{\Theta}_m)}{\sum_k^K \sum_m^M r(\boldsymbol{\Theta}_m) h_k(\boldsymbol{\Theta}_m)}. \quad (3)$$

In the above equations, $\{\boldsymbol{\Theta}_m\}_{m=1}^M$ is the ordered set of the best policy parameters for each component i and for the last L trials, with $r(\boldsymbol{\Theta}_1)h_i(\boldsymbol{\Theta}_1) \geq r(\boldsymbol{\Theta}_2)h_i(\boldsymbol{\Theta}_2) \geq \ldots \geq r(\boldsymbol{\Theta}_M)h_i(\boldsymbol{\Theta}_M)$. M is the parameter of the importance sampler (M can be set to the current number of trials to disable the elite strategy). $\boldsymbol{\Sigma}_0$ is the regularization term avoiding early convergence.

Such GMM representation can cope with two situations:

- When several local optima (separated in the policy parameters space) perform similarly well in terms of rewards. Fig. 2 presents an illustrative view of the problem.
- When the local solution space has a complex, concave or asymmetric shape that cannot be approximated efficiently by a single Gaussian distribution.

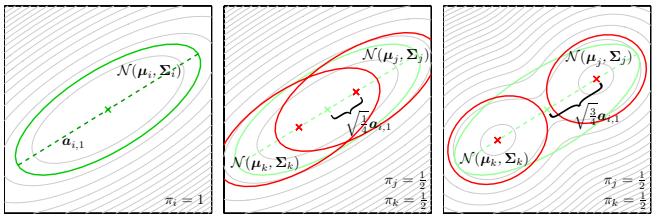

Fig. 3. Gaussian splitting process (in equal proportion). *Left:* Original Gaussian with principal axis $\boldsymbol{a}_{i,1}$ depicted in dashed line and equiprobability lines depicted in light grey color in the background . *Center and right:* Influence of the parameter u on the splitting process ($u = 0$ would correspond to a perfect match of equiprobability lines).

In the first case (Fig. 2), the search for multi-optima policy subspaces can be exploited to evaluate different solutions and select the most appropriate with respect to the current situation (context-dependent selection process). Such choice may for example depend on external factors such as space restriction, occlusion, injured articulation or fatigued muscles (or equivalently, broken or overheated motors). It also allows the agent to robustly adapt to progressively changing environment (slowly drifting reward functions). In this last situation, the system can keep track of regions that might, at a given time, have slightly lower reward, but that remain of interest because they can potentially lead to optimal solutions in the future, or because they can be used as an alternative options if the "best" policy is unavailable or too risky to be used.

The policy space to be explored is initially assumed to be unimodal (a single Gaussian distribution is used, as in PoWER, CEM and CMA-ES). The number of Gaussians is iteratively adapted with the approach described below.

D. GMM split&merge algorithm

Algorithm 1 *GMM split&merge* algorithm.

for $n \leftarrow 1$ to N **do** ▷ Loop for each trial n
  $\boldsymbol{\Theta}_n \leftarrow$ randomSampling($\hat{\Omega}$) ▷ Random sampling of a n-th point
  $\hat{\Omega} \leftarrow$ EM($\hat{\Omega}$,$\boldsymbol{\Theta}$) ▷ Refine policy and exploration noise
  for $i \leftarrow 1$ to K **do** ▷ Compute splitting candidates
    $\Omega_i \leftarrow$ splitGaussian($\hat{\Omega}$,i)
    $L_i^{\mathcal{S}} \leftarrow$ evaluateLikelihood(Ω_i)
  end for
  $\hat{L} \leftarrow$ evaluateLikelihood($\hat{\Omega}$)
  if $(\max(\boldsymbol{L}^{\mathcal{S}}) - \hat{L}) > T^{\mathcal{S}}$ **then**
    $\hat{\Omega} \leftarrow \Omega_{\arg\max(\boldsymbol{L}^{\mathcal{S}})}$ ▷ Split Gaussian (w.r.t. threshold)
  end if
  for $i \leftarrow 1$ to $K-1$ **do** ▷ Compute merging candidates
    for $j \leftarrow i+1$ to K **do**
      $\Omega_{ij} \leftarrow$ mergeGaussians($\hat{\Omega}$,i,j)
      $L_{ij}^{\mathcal{M}} \leftarrow$ evaluateLikelihood(Ω_{ij})
    end for
  end for
  $\hat{L} \leftarrow$ evaluateLikelihood($\hat{\Omega}$)
  if $(\hat{L} - \max(\boldsymbol{L}^{\mathcal{M}})) < T^{\mathcal{M}}$ **then**
    $\hat{\Omega} \leftarrow \Omega_{\arg\max(\boldsymbol{L}^{\mathcal{M}})}$ ▷ Merge Gaussians (w.r.t. threshold)
  end if
end for

We propose to exploit the split&merge mechanism proposed by Zhang *et al* [22] together with the update rule in (3) to adaptively approximate the multi-optima policy solution space. During the learning process, Gaussians are split along their principal axis if the increase of components has a significant impact on the likelihood. Similarly, two Gaussians are merged if the removal of one component only slightly decreases the likelihood.

For a multi-optima policy search, the gradual resolution decrease or increase of the solution space is achieved by merging or splitting the Gaussian distributions and associated *priors* in the current GMM. The merging of two Gaussians j and k into a Gaussian i is characterized by $\pi_i = \pi_j + \pi_k$ and $\pi_i \mathcal{P}(\boldsymbol{\Theta}|i) = \pi_j \mathcal{P}(\boldsymbol{\Theta}|j) + \pi_k \mathcal{P}(\boldsymbol{\Theta}|k)$. It can be shown that the closest result satisfies the relations (see Zhang *et al* [22] for details)

$$\begin{aligned} \pi_i \boldsymbol{\mu}_i &= \pi_j \boldsymbol{\mu}_j + \pi_k \boldsymbol{\mu}_k, \quad \text{and} \\ \pi_i(\boldsymbol{\Sigma}_i + \boldsymbol{\mu}_i \boldsymbol{\mu}_i^\top) &= \pi_j(\boldsymbol{\Sigma}_j + \boldsymbol{\mu}_j \boldsymbol{\mu}_j^\top) + \pi_k(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^\top). \end{aligned} \tag{4}$$

Solving the split equations is an ill-posed problem (there are fewer equations than unknowns), which can however be approximated through *singular value decomposition* (SVD) of the covariance matrices. By defining the set of parameters $l \in \{1, 2, \ldots, N\}$, $\alpha \in [0, 1]$, $\beta \in [0, 1]$ and $u \in [0, 1]$, a Gaussian $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with prior π_i is split into two Gaussians $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ and $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ with parameters

$$\begin{aligned} \pi_j &= \pi_i \alpha, & \pi_k &= \pi_i (1-\alpha), \\ \boldsymbol{\mu}_j &= \boldsymbol{\mu}_i - \sqrt{\frac{\pi_k}{\pi_j}}\, u\, \boldsymbol{a}_{i,l}, & \boldsymbol{\mu}_k &= \boldsymbol{\mu}_i + \sqrt{\frac{\pi_j}{\pi_k}}\, u\, \boldsymbol{a}_{i,l}, \\ \boldsymbol{\Sigma}_j &= \frac{\pi_k}{\pi_j}\boldsymbol{\Sigma}_i + (\beta - \beta u^2 - 1)\frac{\pi_i}{\pi_j}\boldsymbol{a}_{i,l}\boldsymbol{a}_{i,l}^\top + \boldsymbol{a}_{i,l}\boldsymbol{a}_{i,l}^\top, \\ \boldsymbol{\Sigma}_k &= \frac{\pi_j}{\pi_k}\boldsymbol{\Sigma}_i + (\beta u^2 - \beta - u^2)\frac{\pi_i}{\pi_k}\boldsymbol{a}_{i,l}\boldsymbol{a}_{i,l}^\top + \boldsymbol{a}_{i,l}\boldsymbol{a}_{i,l}^\top. \end{aligned} \tag{5}$$

In the above equations, $\boldsymbol{\Sigma}_i = \boldsymbol{A}_i \boldsymbol{A}_i^\top$ represents the ordered eigencomponents decomposition of $\boldsymbol{\Sigma}_i$ with $\boldsymbol{A}_i = [\boldsymbol{a}_{i,1}, \boldsymbol{a}_{i,2}, \ldots, \boldsymbol{a}_{i,D}]$. The parameters $\alpha = \beta = \frac{1}{2}$, $u = \sqrt{\frac{3}{4}}$, $l = 1$ and $D = 1$ are used in our application, corresponding to equally weighted splitting operations by following the principal direction of the Gaussians. Fig. 3 presents an illustration of the split algorithm.

Algorithm 1 presents the pseudocode of the learning process. In the pseudocode, $\hat{\Omega} = \{\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K$ represents the current GMM (initialized with a single Gaussian). Ω_i and Ω_{ij} represent candidate GMMs for the splitting and merging operations. The *randomSampling()* function generates a datapoint $\boldsymbol{\Theta}_n$ from the distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, with i stochastically drawn from $\boldsymbol{\pi}$. The *EM()* function is the expectation-maximization algorithm in (3), used to refine the current GMM $\hat{\Omega}$ to fit the training dataset $\boldsymbol{\Theta}$. The *splitGaussian()* and *mergeGaussians()* functions are defined in (4) and (5). The *evaluateLikelihood()* function computes the average log-likelihood of N weighted datapoints with $\frac{1}{\sum_n^N r(\boldsymbol{\Theta}_n)} \sum_{n=1}^N r(\boldsymbol{\Theta}_n) \log\left(\sum_{k=1}^K \pi_k \mathcal{N}(\boldsymbol{\Theta}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right)$. $T^{\mathcal{S}}$ and $T^{\mathcal{M}}$ are split and merge likelihood thresholds.

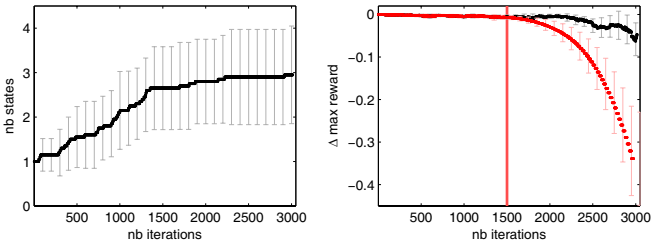

Fig. 4. *Left:* Average number of Gaussians in the GMM at each iteration, with standard deviation depicted as vertical bars. *Right:* Maximum reward at each iteration averaged over 20 runs of the experiment. Each reward is computed by using the centers of the Gaussians with $\hat{r} = \max_i r(\boldsymbol{\mu}_i)$. The simulator provides a controlled environment where the obtained rewards can be compared to the theoretical global optimum computed for each throws accuracy. The black points show 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.

III. DART GAME EXPERIMENT

The learning process is tested in the context of a black-box optimization problem (the policy corresponds to a single action described by two parameters). An agent throws darts at a target and refines its throwing skill while searching for the regions of the dartboard where it should aim to obtain high scores (horizontal and vertical coordinates of the throws). In such configuration, the solution space is progressively transformed from a single to a multi-peaked distribution that needs to be determined to progress in the game. 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.

We assume that the agent refines the precision of its throws linearly in time and and space, which is described by a Gaussian distribution $\mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ with standard deviation σ linearly decreasing from 75 to 16 for 3000 iterations. The average score for each position on the dartboard is computed analytically by exploiting the spectral transformation properties of normal distributions [6]. This provides us with ground-truth data that can be used to evaluate qualitatively and quantitatively the parameters subspaces discovered by the robot.

The experiment is reiterated 20 times, with parameters empirically set to $L = 150$, $M = 15$, $\boldsymbol{\Sigma}_0 = \boldsymbol{I} \cdot 810^{-4}$, $T^{\mathcal{S}} = 0.16$ and $T^{\mathcal{M}} = 0.01$. Fig. 1 presents the results of a typical run. Fig. 4 shows the results averaged over 20 runs, and the effect of stopping the continuous adaptation in the middle of each run. Even though the complexity of the solution space increases with the reduction of the throws variance (from single to multi-peaked distribution, see heatmap in Fig. 1), we can see with the black curve in Fig. 4 that the agent is capable of maintaining a decent score while creating policy alternatives (linear decay trend with low slope). These alternatives can cope with the discontinuous switches of global optima when the throws get more accurate. In contrast, if the agent stops adapting the policy solution space (after 1500 steps) while still improving its throwing capability,

its performance in the game quickly degrades (exponential decay trend of the red curve).

IV. CONCLUSION AND FUTURE WORK

We presented a multi-optima policy exploration approach for continuous learning of skills. The approach relies on a Gaussian mixture model to continuously keep track of possible variants of a policy to enrich the agent's skill and be robust to environmental changes. The approach is tested in a dart game where an agent tries to discover which position on the dartboard maximizes its score while progressively improving its throwing capability. The results showed that the agent could adapt to this slow change of the reward function by continuously driving the exploration/exploitation balance within an expectation-maximization process.

Formulating the search problem as a global fitting of control policy solutions introduces new research perspectives. It first provides additional information about the space of possible solutions. The peaks of the possible solutions are augmented with additional information such as the local spread, shape or curvature of the distribution. Under some conditions, policies with lower peaks and low curvatures may be preferable to policies with higher peaks but high curvatures. The proposed representation provides a framework to explore this research question in future work.

A potential weakness of the current method, that will require further investigation, concerns the threshold parameters that need be set by the experimenter. Future work will study the influence of these parameters on the final learning results, and possible ways of automatically setting them (e.g., with Bayesian information criterion). We plan in future work to compare the incremental *split&merge* solution with other methods for the online estimation of the model structure. In particular, we will investigate the use of Dirichlet processes and spectral clustering to estimate the number of peaks in an online manner.

As a first study, the approach was tested in a dart game experiment formulated as a black-box optimization problem. We plan to study and evaluate in future work how the approach behaves in the case of parameterized policies relying on statistical dynamical systems [16]. We expect that in the case of episodic rewards, the extension to parameterized policy will mostly differ in the dimensionality aspect of the search process (but not in the intrinsic mechanism).

The proposed formulation opens roads for future studies towards the search and selection of multiple policy alternatives. It can for example provide robots with the capability to switch between several policy options in a fast way, without having to relearn the policy (e.g., when policies become unavailable). The formulation also encapsulates additional information about the shape and curvature of the regions of high rewards, which can be exploited to assess how the policy parameters can be disturbed without degrading the skill.

REFERENCES

[1] J. Kodl, G. Ganesh, and E. Burdet, "The CNS stochastically selects motor plan utilizing extrinsic and intrinsic representations," *PLoS ONE*, vol. 6, no. 9, pp. 1–10, 2011.

[2] A. I. T. Salo, I. N. Bezodis, A. M. Batterham, and D. G. Kerwin, "Elite sprinting: are athletes individually step-frequency or step-length reliant?" *Medicine and Science in Sports and Exercise*, vol. 43, no. 6, pp. 1055–1062, 2011.

[3] G. Lawrence, N. Cowan, and S. Russell, "Efficient gradient estimation for motor control learning," in *Proc. of the Intl Conf, on Uncertainty in Artificial Intelligence (UAI)*, 2003, pp. 354–361.

[4] J. Kober, A. Wilhelm, E. Oztop, and J. Peters, "Reinforcement learning to adjust parametrized motor primitives to new situations," *Autonomous Robots*, April 2012.

[5] B. C. da Silva, G. Konidaris, and A. G. Barto, "Learning parameterized skills," in *Proc. Intl Conf. on Machine Learning (ICML)*, 2012.

[6] R. J. Tibshirani, A. Price, and J. Taylor, "A statistician plays darts," *Journal Of The Royal Statistical Society Series A*, vol. 174, no. 1, pp. 213–226, 2011.

[7] J. Peters and S. Schaal, "Using reward-weighted regression for reinforcement learning of task space control," in *Proc. IEEE Intl Symp. on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2007, pp. 262–267.

[8] N. Vlassis, M. Toussaint, G. Kontes, and S. Piperidis, "Learning model-free robot control by a Monte Carlo EM algorithm," *Autonomous Robots*, vol. 27, pp. 123–130, 2009.

[9] T. Rueckstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber, "Exploring parameter space in reinforcement learning," *Paladyn. Journal of Behavioral Robotics*, vol. 1, no. 1, pp. 14–24, 2010.

[10] J. Kober and J. Peters, "Imitation and reinforcement learning: Practical algorithms for motor primitives in robotics," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 55–62, 2010.

[11] P. Dayan and G. E. Hinton, "Using expectation-maximization for reinforcement learning," *Neural Comput.*, vol. 9, no. 2, pp. 271–278, 1997.

[12] S. Schaal and C. Atkeson, "Learning control in robotics: Trajectory-based optimal control techniques," *IEEE Robotics and Automation Magazine*, vol. 17, no. 2, pp. 20–29, 2010.

[13] E. Theodorou, J. Buchli, and S. Schaal, "A generalized path integral control approach to reinforcement learning," *J. Mach. Learn. Res.*, vol. 11, pp. 3137–3181, December 2010.

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

[15] A. Coates, P. Abbeel, and A. Y. Ng, "Apprenticeship learning for helicopter control," *Commun. ACM*, vol. 52, no. 7, pp. 97–105, 2009.

[16] S. Calinon, Z. Li, T. Alizadeh, N. G. Tsagarakis, and D. G. Caldwell, "Statistical dynamical systems for skills acquisition in humanoids," in *Proc. IEEE Intl Conf. on Humanoid Robots (Humanoids)*, Osaka, Japan, 2012.

[17] D. P. Kroese and R. Y. Rubinstein, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer, 2004.

[18] N. Hansen, "The CMA evolution strategy: A comparing review," in *Towards a New Evolutionary Computation*, ser. Studies in Fuzziness and Soft Computing, J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, Eds. Springer Berlin / Heidelberg, 2006, vol. 192, pp. 75–102.

[19] D. Sternad, S.-W. Park, H. Mueller, and N. Hogan, "Coordinate dependence of variability analysis," *PLoS Computational Biology*, vol. 6, no. 4, pp. 1–16, 04 2010.

[20] Z. I. Botev and D. P. Kroese, "The generalized cross entropy method, with applications to probability density estimation," *Methodology and Computing in Applied Probability*, vol. 13, no. 1, pp. 1–27, 2011.

[21] M. Kobilarov, "Cross-entropy motion planning," *Intl Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012.

[22] Z. Zhang, C. Chen, J. Sun, and K. L. Chan, "EM algorithms for Gaussian mixtures with split-and-merge operation," *Pattern Recognition*, vol. 36, no. 9, pp. 1973–1983, 2003.