Local Online Motor Babbling: Learning Motor Abundance of a Musculoskeletal Robot Arm

doi: https://doi.org/10.1109/IROS40897.2019.8967791

Local Online Motor Babbling: Learning Motor Abundance of a Musculoskeletal Robot Arm*

Zinan Liu[1], Arne Hitzmann[2], Shuhei Ikemoto[3], Svenja Stark[1], Jan Peters[1], Koh Hosoda[2]

***Abstract*— Motor babbling and goal babbling has been used for sensorimotor learning of highly redundant systems in soft robotics. Recent works in goal babbling have demonstrated successful learning of inverse kinematics (IK) on such systems, and suggest that babbling in the goal space better resolves motor redundancy by learning as few yet efficient sensorimotor mappings as possible. However, for musculoskeletal robot systems, motor redundancy can provide useful information to explain muscle activation patterns, thus the term motor abundance. In this work, we introduce some simple heuristics to empirically define the unknown goal space, and learn the IK of a 10 DoF musculoskeletal robot arm using directed goal babbling. We then further propose local online motor babbling guided by Covariance Matrix Adaptation Evolution Strategy (CMA-ES), which bootstraps on the goal babbling samples for initialization, such that motor abundance can be queried online for any static goal. Our approach leverages the resolving of redundancies and the efficient guided exploration of motor abundance in two stages of learning, allowing both kinematic accuracy and motor variability at the queried goal. The result shows that local online motor babbling guided by CMA-ES can efficiently explore motor abundance at queried goal positions on a musculoskeletal robot system and gives useful insights in terms of muscle stiffness and synergy.**

I. INTRODUCTION

The human body is an over-actuated system. Not only does it have a higher dimension in motor space than the degrees of freedom (DoF) in the action space, i.e., more muscles than joints, it also has more DoF than necessary to achieve a certain motor task. How the effector redundant system adaptively coordinates movements remains a challenging problem. In the field of robot learning, when assuming rigid body links with pure rotation and translation [1], model learning is commonly used to learn the forward or inverse models of kinematics and dynamics for accurate yet agile control [2]. However, for biomechanical and soft robots such as the elephant trunks [3], or musculoskeletal systems [4] [5], where models based on rigid body links are no longer available, learning becomes difficult due to the highly redundant and non-stationary nature of such systems.

*This project has received funding from the European Union's Horizon 2020 research and innovation program under Grant No 713010 and No 640554, and was supported by JSPS KAKENHI Grant No 18H01410.

[1]Liu Z. is with Department of Computer Science, TU Darmstadt, Hochschulstr. 10 D-64289 Darmstadt, Germany zinan.liu@stud.tu-darmstadt.de

[2]Hitzmann A. is with School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531 Japan arne.hitzmann@arl.sys.es.osaka-u.ac.jp

[3]Ikemoto S. is with Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology, 2-4 Hibikino, Wakamatsu-ku, Kitakyushu 808-0196, Japan ikemoto@brain.kyutech.ac.jp

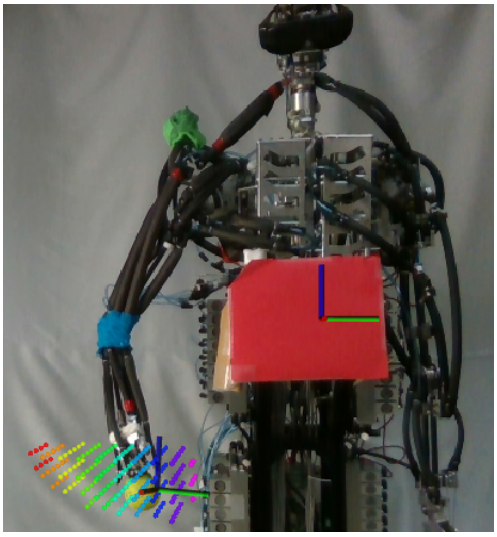


Fig. 1. 10 DoF musculoskeletal robot arm actuated by 24 pneumatic artificial muscles (PAMs), with an empirically defined goal space in reference to the red marker, visualized in `rviz`.

This paper investigates the reaching skills and motor variability of the reached points on a musculoskeletal robot arm [6], an over-actuated system of 24 Pneumatic Artificial Muscles (PAMs) actuating 10 DoFs, as shown in Fig. 1. Traditionally, this problem could be addressed by learning the forward kinematics using motor babbling, and explore the motor-sensory mapping from scratch [7]–[9] until the robot can predict the effects of its actions. However autonomous exploration without prior knowledge in motor babbling doesn't scale well to high dimensional sensorimotor space, due to the rather inefficient sampling of random motor commands in over-actuated systems. An alternative in [10] suggests that learning inverse kinematics with active exploration in goal babbling avoids the curse of dimensionality, simply because the goal space is of much smaller dimension than the redundant motor space. Nonetheless, [10] assumes that the sensorimotor space can be entirely explored, which is not feasible in practice for high dimensional motor systems [3]. Another alternative is then to specify the goal space a priori as a grid, and sampling the goal grid points to guide exploration [11], such that sensorimotor mapping can be sufficiently generalized and bootstrapped for efficient online learning. It has also been quantitatively evaluated for an average of sub-centimeter reaching accuracy on an elephant trunk robot [3] with reasonable experiment time. We therefore implement and further extend on directed goal babbling in [3]. Since the goal space of the robot arm is

unknown and non-convex [6], we empirically estimate it with randomly generated postures, forcing the convex hull such that directed goal babbling can be applied, and subsequently remove the outlier goals in the goal space after learning.

Given the above works aiming to reduce motor redundancy for learning [3], [7]–[11], it can be argued that motor redundancy in human musculoskeletal systems gives rise to the flexible and adaptable natural movements, hence it should be termed motor abundance [12] [13]. In robot motor learning, [14] also suggests that joint redundancy facilitates motor learning, whereas task space variability does not. Thus we leverage the trade-off between goal babbling and motor babbling in two learning stages. Firstly motor redundancy is resolved in the goal babbling phase to accurately learn the IK, and motor babbling guides the exploration using CMA-ES initialized by local samples from goal babbling to "recover" motor abundance. In this way, the exploration in the motor space is effectively constrained to the neighborhood of the queried goal within the goal space, and the explored motor abundance data can be visualized in terms of muscle stiffness and synergies by sampling the fitted Gaussian Mixture Model (GMM).

This paper is organized as follows: in Section II and III directed online goal babbling and CMA-ES are reviewed. Section IV introduces the simple heuristics to define the goal space, implements directed goal babbling on the musculoskeletal robot arm and evaluates the learning results. Section V proposes, implements, and evaluates local online motor babbling using CMA-ES to query motor abundance while providing some insights in muscle stiffness and muscle synergy of the musculoskeletal robot system. Section VI concludes the paper and discusses possible future research.

II. DIRECTED GOAL BABBLING

Given the specified convex goal space $\mathbf{X}^* \in \mathbb{R}^n$ encapsulating K goal points, and denoting all the reachable set of commands in the motor space as $\mathbf{Q} \in \mathbb{R}^m$, the aim is to learn the inverse kinematics model $\mathbf{X}^* \rightarrow \mathbf{Q}$, that generalizes all points in the goal space to a subset of solutions in the motor space. Starting from the known home position x_0^{home}, and home posture q_0^{home}, i.e., the inverse mapping $g(x_0^{home}) = q_0^{home}$, the goal-directed exploration is

$$q_t^* = g(x_t^*, \theta_t) + E_t(x_t^*), \tag{1}$$

where $g(x_t^*, \theta_t)$ is the inverse mapping given learning parameter θ_t, and $E_t(x_t^*)$ adds perturbation noise to discover new positions or more efficient motor commands in reaching goals. At every time step, the motor system forwards the perturbed inverse estimate, $x_t, q_t = \mathrm{fwd}(q_t^*)$, and the actual (x_t, q_t) samples are used for regression, where prototype vectors and local linear mapping [15] is used as the regression model, and to monitor the progress of exploration in the defined goal space.

The major part of directed goal babbling is to direct the babbling of the end-effector at specified goals and target positions. Each trial of goal babbling is directed at one goal randomly chosen from $\mathbf{X}^*$, and continuous piecewise linear targets are interpolated along the path

$$x_{t+1}^* = x_t^* + \frac{\delta_x}{||X_g^* - x_t^*||} \cdot (x_g^* - x_t^*), \tag{2}$$

where x_t^*, X_g^* are the target position and final goal of the trial, and δ_x being the step size. Target positions are generated until x_t^* is closer than δ_x to X_g^*, then a new goal X_{g+1}^* is chosen. The purpose of directed goal babbling is to generate smooth movement around the end-effector position, such that the locally learned prototype vectors can bootstrap and extend the exploration of the goal space, and allow the integration of the following weighting scheme

$$w_t^{\mathrm{dir}} = \frac{1}{2}(1 + \arccos(x_t^* - x_{t-1}^*, x_t - x_{t-1}) \tag{3}$$

$$w_t^{\mathrm{eff}} = ||x_t - x_{t-1}|| \cdot ||q_t - q_{t-1}||^{-1} \tag{4}$$

$$w_t = w_t^{\mathrm{dir}} \cdot w_t^{\mathrm{eff}}. \tag{5}$$

w_t^{dir} and w_t^{eff} measure direction and kinematic efficiency of the movement, such that inconsistency of a folded manifold, and redundant joint positions can be optimized [11]. The multiplicative weighting factor w_t is then integrated to the gradient descent that fits the currently generated samples by reducing the weighted square error.

To prevent drifting to irrelevant regions and facilitate bootstrapping on the local prototype centers, the system returns to (x^{home}, q^{home}) with probability p^{home} instead of following another goal directed movement. Returning to home posture stablizes the exploration in the known area of the sensorimotor space [12], [18], similar to infants returning their arms to a comfortable resting posture between practices:

$$q_{t+1}^* = q_t^* + \frac{\delta_q}{||q_{home} - q_t^*|| \cdot (q^{home} - q_t^*)}. \tag{6}$$

The system moves from the last posture q_t^* to the home posture q^{home} in the same way as in (2) by linearly interpolating the via-points along the path, until $||q_{home} - q_t^*|| < \delta_q$.

The exploratory noise, or motor perturbation in (1), is crucial for discovering new postures that would otherwise not be found by the inverse estimate [12], [29]. By initializing each motor dimension with a normal distribution of zero mean and certain variance, the following motor commands can be sampled with the perturbation noise, which is also drawn from a separate normal distribution.

$$E_t(x_t^*) = A_t \cdot x_t^* + b_t, \quad A_t \in \mathbb{R}^{m \times n}, \quad b_t \in \mathbb{R}^m, \tag{7}$$

where all entries e_t^i in the matrix A_t is initialized and varied

$$e_0^i \sim \mathcal{N}(0, \sigma^2), \quad \delta_{t+1}^i \sim \mathcal{N}(0, \sigma_\Delta^2)$$

$$e_{t+1}^i = \sqrt{\frac{\sigma^2}{\sigma^2 + \sigma_\Delta^2}} \cdot (e_t^i + \delta_{t+1}^i) \sim \mathcal{N}(0, \sigma^2).$$

In this way the local surrounding of the end-effector can be well explored, whereas the variance is scaled by the sum of the added variance to avoid sudden jumps in the exploration.

After learning, the average reaching accuracy is evaluated by querying the inverse model for every goal within the

defined goal space $\mathbf{X}^*$, and a simple feedback controller to adapt to execution failures. Execution failure occurs when the inverse estimate is not possible to execute, i.e., $q^* \notin \mathbf{Q}$, due to interference and non-stationary changes in $\mathbf{Q}$. Thus a simple feedback controller is introduced [3]

$$\hat{x}_0^* = x^*, \quad \hat{x_{t+1}^*} = \hat{x_t^*} + \alpha \cdot \mathrm{err}_t. \tag{8}$$

Given the queried goal x^* and the predicted posture $q^* = g(x^*)$, where $q^* \notin \mathbf{Q}$, the feedback controller would slightly shift the queried goal from x^* to $\hat{x}_t^*$, proportional to the observed error $\mathrm{err}_t = x^* - x_t$ integrated over time, then forwarding the inverse estimate $x_t = \mathrm{fwd}(g(\hat{x}_t^*))$.

III. CMA-ES

CMA-ES is a method of black box optimization that minimizes the objective function $f : \mathbf{Q} \in \mathbb{R}^m \to \mathbb{R}$, $q \to f(q)$, where f is assumed to be a high dimensional, non-convex, non-separable, and ill-conditioned mapping of the multi-variate state space. The idea of CMA-ES is introducing a multi-variate normal distribution to sample a population, evaluating the population $f(\mathbf{q})$ to select the good candidates, and updating the search distribution parameters by adapting the covariance and shifting the mean of the distribution according to the candidates.

Given a start point q^0 and initializing the covariance to identity matrix $\mathbf{C^0} = \mathbf{I}$, the search points in one population iteration is sampled as follows:

$$q_i^t \sim m^t + \sigma^t y_i^t \quad i = 1, \cdots, \lambda \quad q_i, m \in \mathbb{R}^n, \sigma \in \mathbb{R}_+, \mathbf{C} \in \mathbb{R}^{n \times n} \tag{9}$$

where $y_i^t = \mathcal{N}_i(\mathbf{0}, \mathbf{C^t})$, m being the mean vector, σ being the step-size, and λ is the population size. For notation simplicity, the iteration index t is henceforth omitted.

The mean vector m is updated by using the non-elitistic selection [16]. Let $q_{i:\lambda}$ denote the ith best solution in the population of λ, the best μ points from the sampled population are then selected, such that $f(q_{1:\lambda}) \leq \cdots \leq f(q_{\mu:\lambda})$, and weighted intermediate recombination is applied:

$$m \leftarrow m + \Sigma_{i=1}^{\mu} w_i y_{i:\lambda} =: m + y_w, \tag{10}$$

$$\text{where } w_1 \geq \cdots \geq w_\mu > 0, \ \Sigma_{i=1}^{\mu} w_i = 1, \ \frac{1}{\Sigma_{i=1}^{u} w_i^2} =: \mu_w \approx \frac{\lambda}{4}.$$

The step size σ is updated using cumulative step-size adaptation (CSA). The intuition is when the evolution path, i.e., the sum of successive steps, is short, single steps tend to be uncorrelated and cancel each other out, thus the step-size should be decreased. On the contrary, when evolution path is long, single steps points to similar directions and tend to be correlated, therefore increasing the step size. Initializing the evolution path vector $p_\sigma = \mathbf{0}$, and setting the constants $c_\sigma \approx 4/n, d_\sigma \approx 1$, the step size is updated as:

$$p_\sigma \leftarrow (1 - c_\sigma) p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_w}\, y_w \tag{11}$$

$$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma} \left(\frac{||p_\sigma||}{E||\mathcal{N}((\mathbf{0}, \mathbf{I}))||} - 1 \right) \right) \tag{12}$$

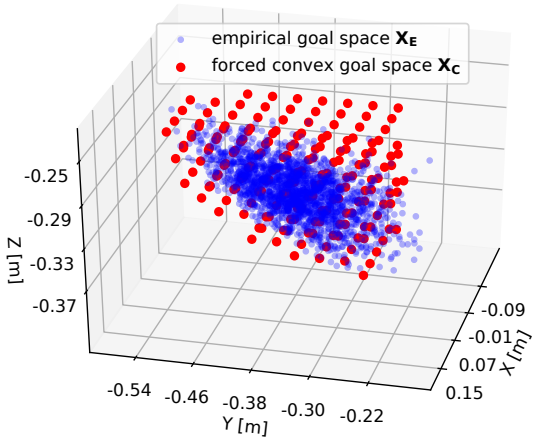


Fig. 2. Empirical goal space $\mathbf{X_E}$(in blue) sampled from 2000 random postures, and the convex goal space $\mathbf{X_C}$(in red), which is used for learning as shown in Fig. 1

The essential part of the evolution strategy is the covariance matrix adaptation. It is suggested that the line distribution adapted using rank-one update will increase the likelihood of generating successful steps $\mathbf{y_w}$, because the adaptation follows a natural gradient approximation of the expected fitness of the population $f(\mathbf{q})$

$$p_c \leftarrow (1 - c_c) p_c + \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_w}\, y_w \tag{13}$$

$$\mathbf{C} \leftarrow (1 - c_{cov})\mathbf{C} + c_{cov} p_c p_c^T \tag{14}$$

IV. INVERSE KINEMATICS LEARNING

We use a 10 DoF musculoskeletal robot arm from [6] for the experiments. The arm is driven by 24 pneumatic muscles, each with pressure actuation range of $[0, 0.4]$MPa. As shown in Fig. 1, the hand of the robot is replaced with a tennis ball as the color marker, and tracking of the end-effector is performed in reference to the center of the red marker as the origin, using Intel RealSense ZR300. However the tracking introduces an error up to 1cm in depth, i.e., x-axis, and sub-millimeter error in y and z axis. The colored point cloud overlayed in ROS rviz is the specified convex goal space as in Fig. 1. The control accuracy of the robot is tested according to [3]. By repeating $P = 20$ random postures for $R = 20$ times each, the average Euclidean norm error is computed as $\bar{x}_p = \frac{1}{R}\sum_r x_p^r$, where $D = \frac{1}{P}\sum_p \frac{1}{R}\sum_r ||x_p^r - \bar{x}_p||$, and $D = 1.2$cm.

A. Define the Goal Space

The complete task space of the upper limb robot is unknown and non-convex, however, directed goal babbling would require the specified goal space to be convex to efficiently bootstrap and allow the integration of the weighting scheme in (5). Thus we first empirically estimate the goal space by randomly generating 2000 random postures for each muscle within $[0, 0.4]$ MPa, and take the encapsulated convex hull as the empirical goal space X_E. In order to approximate the uniform samples in X_E for efficient online learning and

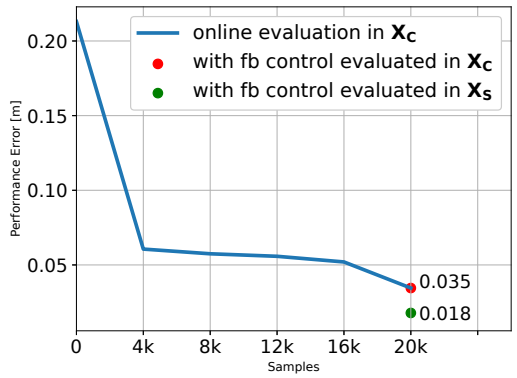


Fig. 3. The average Euclidean norm error to all the goals in the convex goal space $\mathbf{X_C}$, evaluated online after every 4000 samples. The feedback controller is applied at 20000 samples, showing an average error of 3.5 cm (red dot). However there are still outlier goals remaining from forcing the convex hull, thus we take the explored prototype sphere space $\mathbf{S}$ and intersect with $\mathbf{X_C}$, i.e., $\mathbf{X_S} = \mathbf{S} \cap \mathbf{X_C}$ to remove the outliers. Evaluating on the cut goal space $\mathbf{X_S}$ reduces the error to 1.8 cm (green dot)

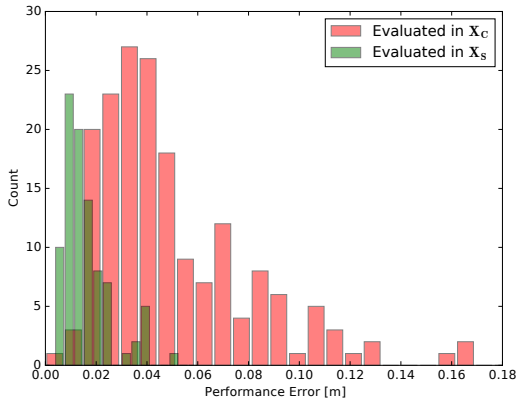


Fig. 4. Performance error distribution in red bars are evaluated in goal space $\mathbf{X_C}$ in Fig. 2 used for IK learning, and the one in green bars are evaluated in the post-processed goal space $\mathbf{X_S}$ in Fig. 5

evaluations, a cube grid $\mathbf{C}$ with $3cm$ spacing encapsulating $\mathbf{X}_E$ is defined, where $\mathbf{X}_E \subset \mathbf{C}$. The sampled convex hull goal grid $\mathbf{X_C}$ in Fig. 1 is then made from the intersection of all points in the empirical goal space and the cube grid, i.e., $\mathbf{X_C} = \mathbf{X}_E \cap \mathbf{C}$. However, as shown in Fig. 2, $\mathbf{X}_E$ is a slanted non-convex irregular ellipsoid, forcing a convex hull in the empirical goal space would introduce non-reachable regions. This is addressed later with a similar set operation to remove the outlier goals using the learned prototype vectors.

B. Experiment and Results

The experiment is conducted with $T = 20000$ samples, with target step length $\delta_x = 0.02$, which corresponds to the target velocity of 2 cm/s, allowing the robot to generate smooth local movements. The sampling rate is set to $5Hz$, generating 5 targets and directed micro-movements for learning. After every 4000 samples are generated, performance error is evaluated online by querying every goal in $\mathbf{X_C}$, executing the estimated IK command, and compute the average euclidean distance to the goal. The learning experiment including online evaluations amounts to less than

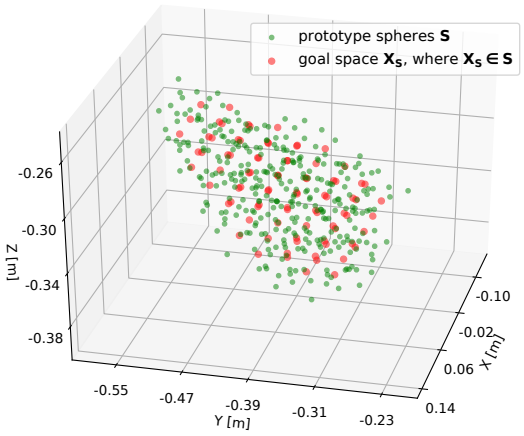


Fig. 5. Prototype spheres $\mathbf{S}$ (green) encapsulating the final goal space $\mathbf{X_S}$ (red), after outlier goals have been removed.

2 hours real-time. As illustrated in Fig. 3, in the first 4000 samples the performance error greatly reduces, meaning a fast bootstrapping of local models that spread throughout the $\mathbf{X_C}$. The rest of the learning is followed by some finer guided exploration in $\mathbf{X_C}$, which is relatively well-explored in the first 4000 samples. At $T = 20000$, the feedback controller is applied, where the performance error drops to 3.4cm. However in $\mathbf{X_C}$ there are still many outlier goals, which are the non-reachable regions introduced by forcing the convex hull. A similar set intersection operation is applied to remove outlier goals, i.e., $\mathbf{X_S} = \mathbf{S} \cap \mathbf{X_C}$, where $\mathbf{S}$ is taken as the encapsulated space of the learned prototype spheres, as shown in Fig. 5. We then evaluate again these goals with the feedback controller, the performance error reduces further to an average of 1.8 cm in Fig. 3. However, due to the forced convex hull X_C, local inverse models cannot efficiently regress at the edge of the task space, the error distribution still shows a few errors larger than 3 cm, which can be further reduced later by motor babbling using CMA-ES.

V. LEARNING MOTOR ABUNDANCE

CMA-ES explores by expanding the search distribution of the parameters, shifting the mean and expanding the covariance, until an optimum solution is found within that distribution, followed by shrinking the covariance and shifting the mean to the global optimum. By intentionally setting the initial mean vector slightly away from the optimum, i.e., the posture that leads to closest end-effector position to the goal, CMA-ES would naturally expand the covariance while keeping the search within the vicinity of the queried goal, as the objective function is set to minimize the goal-reaching error. Essentially CMA-ES is used to effectively generate motor babbling data, which can be achieved by initializing the mean vector with neighboring goal postures $g(\hat{x})$ of the queried goal x^*, and the step-size with the empirical variance estimate of the local samples around x^* gathered from the goal babbling process.

A. Local Online Motor Babbling

When learning inverse kinematics using online goal babbling, since there are multiple postures $\mathbf{q}$ reaching x^*, it is assumed that we don't need to know all redundancy, and only learn the ones with most direction and kinematic efficiency by integrating the weighting scheme (5) in the optimization. In fact, $\mathbf{Q}$ is not only unknown, and may never be exhaustively explored on a physical system, but also non-stationary due to the nature of musculoskeletal robot design with PAMs. This can be addressed by using the simple feedback controller in (8), where execution failures due to the changing of $\mathbf{Q}$ are adapted when the queried goal x^* is slightly shifted based on the observed error.

Algorithm 1: Motor Babbling Using CMA-ES

input : x^*, $g(x)$, $\mathbf{Q}_{\hat{\mathbf{x}}}$
output : $\mathbf{Q}_{\mathbf{cma}}$
initialize: $\alpha = 0.05$, $T = 30$, $N = 5$, $\lambda = 13$, $r = 0.02$, $c = 10$, $f^* = 0.03$, $\mathbf{Q}_{\mathbf{cma}} = \{\}$

select N closest goals $x_1^*, \cdots, x_N^*$ *to* x^*;
for $n \leftarrow 0$ **to** N **do**
  $\hat{x}_0^* = x_n^*$;
  $\mathbf{Q}_{\mathbf{fb}} = \{\}$;
  for $t \leftarrow 0$ **to** T **do**
    $x_t, q_t = \text{forward}(g(\hat{x}_t^*))$;
    $\hat{x}_t^* = \hat{x}_{t-1}^* + \alpha \cdot (x^* - x_t)$;
    if $||x^* - x_t|| < r$ **then**
      collect $(x_t,\ q_t)$ In $\mathbf{Q}_{\mathbf{fb}}$;
    end
  end
  select q_t *for the minimum* $||x_t - x^*||$ *in* $\mathbf{Q}_{\mathbf{fb}}$;
  initialize $m = q_t$, $\sigma = mean(var(\mathbf{Q}_{\hat{\mathbf{x}}} \cup \mathbf{Q}_{\mathbf{fb}}))$, $\mathbf{C} = \mathbf{I}$;
  while $\hat{f} < f^*$ **do**
    sample posture population $\mathbf{q_s} : q_1 \cdots q_\lambda$ as in (9);
    for $k \leftarrow 1$ **to** λ **do**
      $x_t, q_t = \text{forward}(q_k)$;
      $\hat{f} = f(x_k) = c \cdot ||x^* - x_t||$;
      if $||x^* - x_t|| < r$ **then**
        collect q_k in $\mathbf{Q}_{\mathbf{cma}}$;
      end
    end
    update $\mathbf{m}$ as in (10);
    update $\mathbf{p}_\sigma$ and σ as in (11), (12);
    update $\mathbf{p_c}$ and $\mathbf{C}$ as in (13), (14);
  end
end

As illustrated in Algorithm 1, the queried goal x^*, the learned inverse model $g(x)$, and the neighboring postures $\mathbf{Q}_{\hat{\mathbf{x}}} : q_t \forall x_t \iff ||x_t - x^*|| < r$, which are collected from the goal babbling process, are the input to online motor babbling. The aim of the algorithm is to output a new posture configuration set $\mathbf{Q}_{\mathbf{cma}}$, from which different muscle stiffness can be generated while keeping the end-effector position fixed. The initialization sets the gain and number of iteration of the feedback controller to $\alpha = 0.05$, $T = 30$, t number of trials for CMA-ES $N = 5$, and the prototype sphere radius is $r = 0.02$. We use `pycma` library [17] to implement CMA-ES, where we encode variables q in the objective function implicitly $f(\text{fwd}(q))$ [16]. The objective function is simply set as the euclidean norm to the goal scaled with a constant, i.e., $c \cdot ||x^* - x_t||$, where $c = 10$, and the optimum objective function value is set to $f^* = 0.03$, meaning that an empirical optimum of $f^*/c = 3mm$ to the goal, which is also the stopping criteria for each CMA-ES trial.

Each trial of CMA-ES starts by iterating the feedback controller and finding the posture q_t that leads closest to the neighboring goal, and q_t is subsequently used to initialize the mean vector m. The covariance is initialized to be an identity matrix, which allows isotropic search and avoids bias. In order to initialize the step-size, an empirical variance is estimated from $\mathbf{Q}_{\hat{\mathbf{x}}} \cup \mathbf{Q}_{\mathbf{fb}}$, and the mean of the variance is taken as initialization. The union of the two sets is to ensure sufficient data for a feasible estimation. Near the home position, which is the centroid of the goal space, many data samples are available as online goal babbling often comes back to (x_{home}, q_{home}). However, around the edges of the goal space, there are often very few local samples, sometimes less than the action space dimension, i.e., the 24 muscles. By taking in the samples generated by the feedback controller, a better initialization of σ can be robustly estimated.

B. Visualizing Muscle Abundance

In order to visualize muscle abundance, namely in terms of reproducing muscle stiffness and muscle synergy encoded in the evolved covariance matrix, we assume the distribution of parameters to be multi-variate Gaussian and multi-modal, as the motor space is of high dimension, and there can be different muscle group posture configurations while keeping the end-effector fixed. Therefore a multi-variate Gaussian Mixture Model [18] is fit to the collected data in $\mathbf{Q}$. By assuming a distribution of Gaussian parameters over the data samples $p(\mathbf{Q}|\theta)$, a prior multi-variate Gaussian distribution is introduced as $p(\theta) = \Sigma_{i=1}^{K} w_i \mathcal{N}(\mu_\mathbf{i}, \Sigma_i)$, where w_i are the weights for each Gaussian mixture component. The posterior distribution is estimated by using Bayes rule [18], such that the posterior distribution would preserve the form Gaussian mixture model, i.e., $p(\theta|\mathbf{Q}) = \Sigma_{i=1}^{K} \tilde{w}_i \mathcal{N}(\tilde{\mu}_\mathbf{i}, \tilde{\Sigma}_i)$, where the parameters $(\tilde{\mu}_\mathbf{i}, \tilde{\Sigma}_i)$ and weights $\tilde{w}_i$ are updated using Expectation Maximization (EM) to maximize the likelihood [18]. The number of mixture models P is estimated using Bayesian Information Criterion (BIC) [18] for $P \in [1, 10]$, where the lowest BIC of P is taken. Finally, we sample from the mixture model with updated parameters and weights $q^* \sim \Sigma_{i=1}^{K} \tilde{w}_i \mathcal{N}(\tilde{\mu}_\mathbf{i}, \tilde{\Sigma}_i)$ and forward q^* on the robot.

C. Experiment and Results

We evenly selected 10 goals in the final goal space $\mathbf{X_S}$ to perform online motor babbling. The selected goals and their local samples within the 2cm radius are shown in Figure 6. The goals are selected to showcase the generality of querying any goal within the goal space for motor babbling. Around

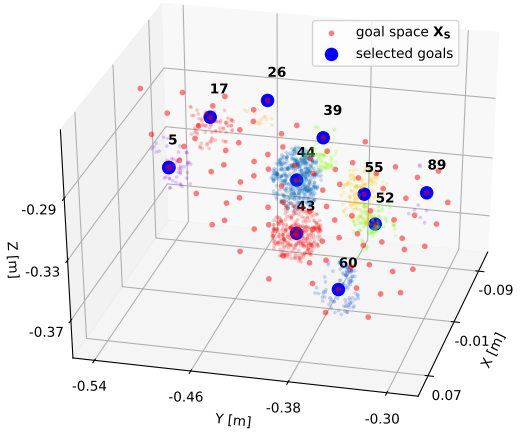


Fig. 6. 10 selected goals for motor babbling in the final goal space $\mathbf{X_S}$, the color point clouds are the local samples within 2cm radius of the queried goals, which are used to initialize the step-size σ for the CMA-ES trials

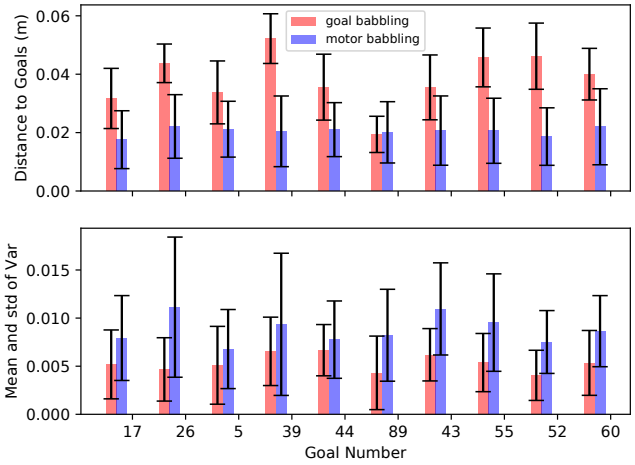


Fig. 7. Comparing the reaching error and muscle variability of directed goal babbling and local motor babbling using CMA-ES, where CMA-ES not only increased the means and standard deviations of all 24 muscle variances for the 10 queried goals, but the reaching error has also been reduced

the edges, goal 26, 5, 89, 52 are chosen, and near the centroid home position, goal 44 and 39 are selected. The remaining goals 17, 43, 55, and 60 are to populate the rest of the goal space. It can be expected that more samples were generated near the home posture, since in online goal babbling the arm returns to (x_{home}, q_{home}) with probability p_{home}, whereas goals around the edges have only a few samples, such as goal 26 and 89.

For each selected goal, $N = 5$ trials of CMA-ES is performed as in Algorithm 1, where each trial takes on average 5 minutes experiment time on the robot. Muscle stiffness is then reproduced by first fitting the collected neighboring samples $\mathbf{Q}_{\hat{\mathbf{x}}}$ to the Gaussian mixture model, which serves as a baseline learned during goal babbling, followed by another experiment fitting $\mathbf{Q}_{\mathbf{cma}}$ to the mixture model and the subsequent sampling. 200 samples from the mixture model are evaluated on the robot, the mean and standard

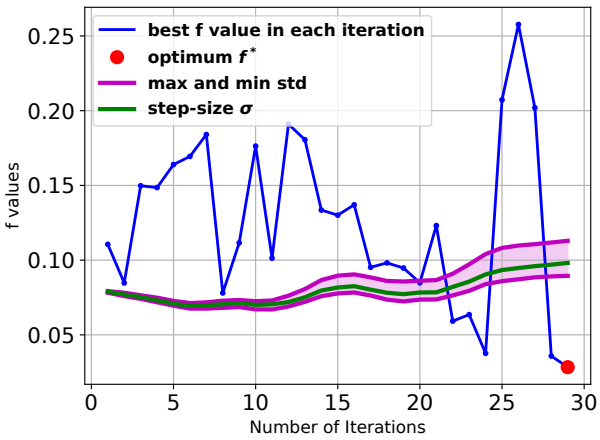


Fig. 8. One evolution trial for goal 44, the search of the step-size increases until the defined optimum objective function value is found

deviation of the reaching error, and of the pressure variance are plotted. As illustrated in Fig. 7, CMA-ES outperforms the baseline in terms of both larger muscle pressure variance and smaller goal-reaching error, where the average lies close to the 2cm prototype sphere radius. Since online goal babbling favors kinematic and direction efficiency by reducing motor redundancy, the sampled muscle pressure generally varies trivially compared to the ones generated from the CMA-ES GMM model, which expands the variance in search of global optimum while keeping the goal-reaching accuracy. Due to non-stationary changes of the possible posture configurations $\mathbf{Q}$, the local neighboring samples $\mathbf{Q}_{\hat{\mathbf{x}}}$ no longer lead to a close position to the goal, however $\mathbf{Q}_{\hat{\mathbf{x}}}$ of the neighboring goals can be used for initializing the step-size σ, and initializing the mean vector m from $\mathbf{Q}_{\mathbf{fb}}$, to adapt to non-stationary changes.

It can be observed that for goal 44, which is closest to the home position, the motor variance doesn't increase as much as other queried goals. This is because every time the interpolated directed goal path comes across the centroid home region, goal 44 has a higher chance of collecting more samples q_t of varied motor configurations within the neighborhood. Nevertheless, CMA-ES still explores motor redundancy rather efficiently. As shown in Fig. 8, the evolution trial expands the maximum and minimum standard deviation of the search, i.e., such that the optimum f^* is reached. After 5 such evolution trials, the sampled GMM data is used to estimate the covariance, compared with the covariance estimate from the baseline GMM data. As shown in 9, CMA-ES preserves the structure while enhancing the variance on the diagonal, while also discovers more correlation within different groups of muscles, which can be prominently observed on the robot in Fig. 10.

D. Interpreting Muscle Abundance

The muscle pressure variability in the covariance encodes muscle abundance, which can be interpreted as muscle stiffness and static muscle synergies. Loosely speaking, muscle synergy is defined as a co-activation pattern of muscles in a certain movement from a single neural command signal

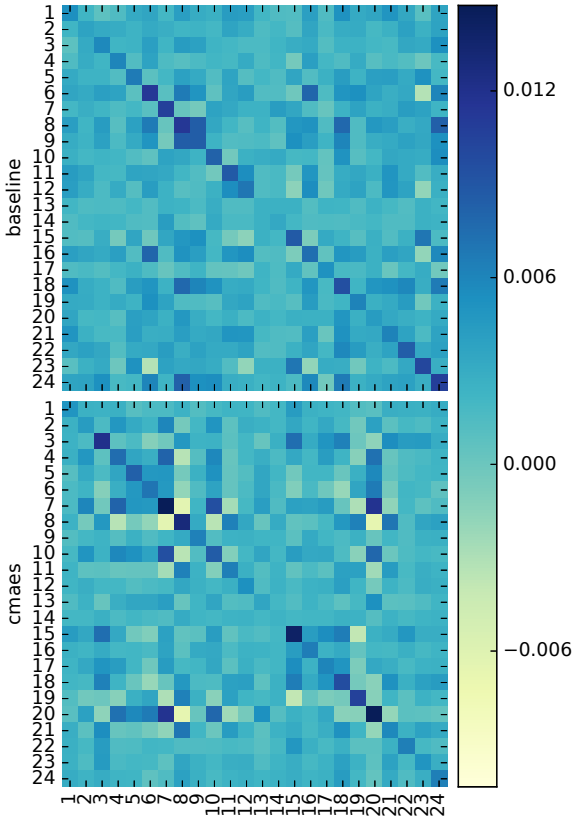
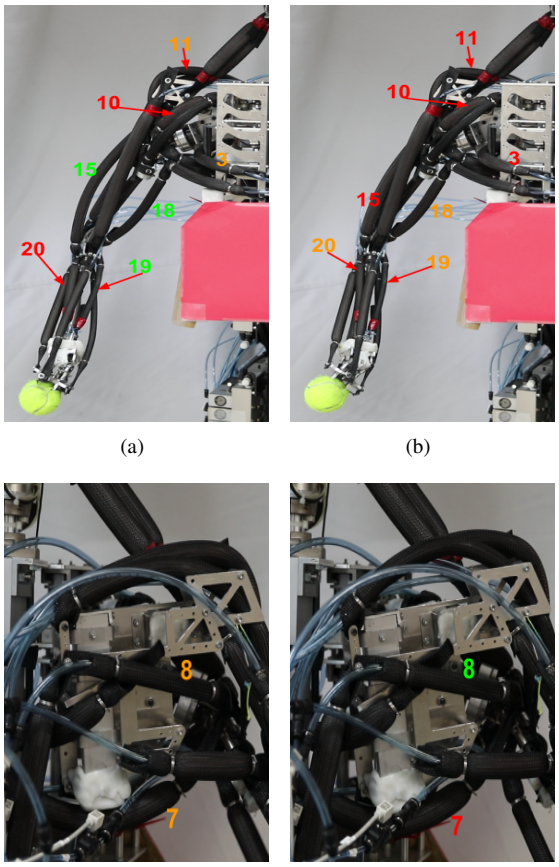


Fig. 9. Comparing baseline and CMA-ES covariances, where the base structure is preserved yet with enhanced correlations. The largest change of variance occurs at muscle pair (8,20) from 0.003 to -0.01, where the -0.01 covariance corresponds to the standard deviation of 0.1 MPa pressure change, consituting 25% of the PAM actuation range

#	Muscle Name	Function
3	serratus anterior	pulls scapula forward
7	latissimus dorsi	rotates scapula downward
8	rear deltoid	abducts, flexes,
10	front deltoid	and extends
11	medial deltoid	the shoulder
15	biceps brachii	flexes and supinates the forearm
18	brachialis	flexes the elbow
19	pronator	pronates the hand
20	supinator	supinates the hand

TABLE I
MUSCLE NAMES AND FUNCTIONS

[19]. It can be argued that muscle synergy is a way of kinetically constraining the redundant motor control of limited DoFs, or as neural strategies to handle the sensorimotor systems [20]. By constraining the end-effector position of the musculoskeletal robot arm, the static muscle synergies and stiffness can be encoded in the covariance matrix and provide some useful insights. In Fig. 9, muscles of high variances, namely muscle 3, 7, 8, 10, 11, 15, 18, 19, 20 are of particular interest, where muscle 7 and 8, 20 and 8 are highly negatively correlated. Inspired by the human's upper limb, the PAMs of the robot arm mimics the function of human arm muscles, as illustrated in Table I. By fitting the data $\mathbf{Q_{cma}}$ in the mixture models and subsequently applying sampling, we can observe the co-activation patterns of the



(a) (b) (c) (d)

Fig. 10. Motor abundance replay by sampling the fitted GMMs: fixed end-effector position with varied muscle pressures gives rise to static muscle synergies. The labeled muscles are color-coded in green (low), orange (medium), and red (high) to indicate the state of pressure actuation. A relaxed arm posture with a lowered shoulder can be observed in (a) and (c), whereas a stiffened arm with a pronating hand and a lifted shoulder and can be observed in (b) and (d), while keeping the end-effector position fixed.

muscles. As shown in Fig.10(a) and 10(c), the upper limb first reaches goal 44 with a relaxed arm posture and a lowered adducted shoulder, whereas in Fig.10(b) and 10(d) the end-effector position is maintained by stiffening the arm, lifting the extended shoulder, and pronating the hand. The negative correlation of muscle 7 and 8 can be interpreted as the coordination of extension and abduction, as well as the flexion and adduction of the shoulder. Muscle 8 and 20 coordinate shoulder abduction with a supinating hand, and by adducting the shoulder while pronating the hand.

VI. CONCLUSIONS

We have implemented directed goal babbling [11] to learn the inverse kinematics of a 10 DoF musculoskeletal robot arm actuated by 24 PAMs. We empirically estimated and forced the convex goal space, followed by post-processing to remove outlier goals. The learning result shows an average reaching error of 1.8 cm, where the reaching accuracy achievable by the robot is 1.2 cm. The simple heuristics and approximation of the goal space allow us to use directed goal babbling to learn the IK in an unknown goal space.

Nevertheless, learning with a forced convex goal space where the intrinsic task space is non-convex introduces outlier goals, which leads to relatively large reaching errors along the non-convex edge. A future research direction of integrating directed goal babbling with active exploration could be of interest [10], where the goal space grid can be defined large enough to encapsulate the whole task space, and active exploration guided by the k-d tree splitting and progress logging can indicate the learned task space while still keeping the bootstrapping flavor of the local learners.

We further extended directed goal babbling to local online motor babbling using CMA-ES in search of more motor abundance. By initializing the evolution strategy with local samples generated from goal babbling, any point within the goal space can be queried for motor abundance. The idea is to intentionally initialize the mean vector of CMA-ES slightly away from the queried goal. By expanding the covariance and setting the stop condition to meet the set optimum of the objective function value, efficient motor babbling data can be generated locally around the queried goal with a few CMA-ES trials of different initializations from the neighboring goals. We evenly selected 10 goals throughout the goal space to showcase the generality of local online motor babbling. The results show that our proposed method has significantly increased the average muscle variability compared to the goal babbling baseline, while keeping the end-effector more stable. The collected motor abundance data can be fit to Gaussian mixture models, and the sampling of the GMMs can be used to reproduce motor abundance in terms of muscle stiffness and muscle synergies encoded in the evolved covariance matrix. The bonus that comes with the fitted GMMs is that the queried motor abundance can be captured and reproduced by distributions, which enables the formulation of reinforcement learning trials in future research, such as learning weight lifting with varied muscle stiffness, trajectory planning by sampling motor commands at the via-points to collect demonstrations and possibly integrate with the probabilistic movement primitives [21].

Another future research direction would be to investigate the motor abundance of continuum-based soft robotic systems. When precise end-effector manipulation with flexible motor adaptation is needed, exploring the motor abundance could be a good start. Given that accurate IK can be learned with such systems [3], local online motor babbling that bootstraps on goal babbling would also scale well to effector redundant systems with infinitely many DoFs. The challenge, however, lies in the initialization of the starting positions of CMA-ES. For musculoskeletal systems, it can be initialized by random neighboring goals or heuristically selected ones, since motor variability is constrained by the skeletal systems and appears rather intuitive. For continuum soft bodies, initializing with prior knowledge might be needed for efficient motor babbling, such as the neighboring goals of higher sample variance from the goal babbling stage.

ACKNOWLEDGMENT

The author would like to thank Hiroaki Masuda for his mechanical maintenance of the upper limb robot, and Dr. Matthias Rolf for his suggestions in implementing directed online goal babbling.

REFERENCES

[1] R. P. Paul, *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. Richard Paul, 1981.

[2] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, 2011.

[3] M. Rolf and J. J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 6, 2014.

[4] K. Hosoda, S. Sekimoto, Y. Nishigori, S. Takamuku, and S. Ikemoto, "Anthropomorphic muscular–skeletal robotic upper limb for understanding embodied intelligence," *Advanced Robotics*, vol. 26, no. 7, 2012.

[5] R. Niiyama, S. Nishikawa, and Y. Kuniyoshi, "Biomechanical approach to open-loop bipedal running with a musculoskeletal athlete robot," *Advanced Robotics*, vol. 26, no. 3-4, 2012.

[6] A. Hitzmann, H. Masuda, S. Ikemoto, and K. Hosoda, "Anthropomorphic musculoskeletal 10 degrees-of-freedom robot arm driven by pneumatic artificial muscles," *Advanced Robotics*, vol. 32, no. 15, 2018.

[7] P. Gaudiano and S. Grossberg, "Vector associative maps: Unsupervised real-time error-based learning and control of movement trajectories," *Neural networks*, vol. 4, no. 2, 1991.

[8] Y. Demiris and A. Dearden, "From motor babbling to hierarchical learning by imitation: a robot developmental pathway," 2005.

[9] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001.

[10] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, 2013.

[11] M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *Development and Learning (ICDL), 2011 IEEE International Conference on*, vol. 2. IEEE, 2011.

[12] M. Latash, "There is no motor redundancy in human movements. there is motor abundance," 2000.

[13] M. L. Latash, "The bliss (not the problem) of motor abundance (not redundancy)," *Experimental brain research*, vol. 217, no. 1, 2012.

[14] P. Singh, S. Jana, A. Ghosal, and A. Murthy, "Exploration of joint redundancy but not task space variability facilitates supervised motor learning," *Proceedings of the National Academy of Sciences*, vol. 113, no. 50, 2016.

[15] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, 1990.

[16] N. Hansen, "The cma evolution strategy: A tutorial," *arXiv preprint arXiv:1604.00772*, 2016.

[17] N. Hansen, Y. Akimoto, and P. Baudis, "CMA-ES/pycma on Github," Zenodo, DOI:10.5281/zenodo.2559634, Feb. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2559634

[18] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[19] G. Torres-Oviedo, J. M. Macpherson, and L. H. Ting, "Muscle synergy organization is robust across a variety of postural perturbations," *Journal of neurophysiology*, 2006.

[20] M. C. Tresch and A. Jarc, "The case for and against muscle synergies," *Current opinion in neurobiology*, vol. 19, no. 6, 2009.

[21] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann, "Probabilistic movement primitives," in *Advances in neural information processing systems*, 2013.