Efficient Exploratory Learning of Inverse Kinematics on a Bionic Elephant Trunk

Matthias Rolf and Jochen J. Steil

***Abstract*— We present an approach to learn the inverse kinematics of the "bionic handling assistant"—an elephant trunk robot. This task comprises substantial challenges including high dimensionality, restrictive and unknown actuation ranges, and nonstationary system behavior. We use a recent exploration scheme, online goal babbling, which deals with these challenges by bootstrapping and adapting the inverse kinematics on the fly. We show the success of the method in extensive real-world experiments on the nonstationary robot, including a novel combination of learning and traditional feedback control. Simulations further investigate the impact of nonstationary actuation ranges, drifting sensors, and morphological changes. The experiments provide the first substantial quantitative real-world evidence for the success of goal-directed bootstrapping schemes, moreover with the challenge of nonstationary system behavior. We thereby provide the first functioning control concept for this challenging robot platform.**

***Index Terms*— Bionic handling assistant (BHA), continuum robot, goal babbling, inverse kinematics.**

I. Introduction

MOTOR learning is an important application of machine learning and increasingly relevant for modern robotics. Already standard robots with well-known geometry and mass distribution largely benefit from learning for the purpose of accurate and agile motor control [1]. Learning is even more important for new generations of robots that combine mechanical flexibility, elastic material, and lightweight actuation-like pneumatics. Such robots are often inspired by biological actuators-like octopus arms [2], elephant trunks [3], or human biomechanics [4], and provide enormous potential for the physical interaction between the robot and the world, and in particular between robots and humans. The downside of their biologically inspired design is that analytic models for their control are hardly available, which qualifies learning as an essential tool for their successful application. Robots with elastic elements face additional problems with nonstationary behaviors due to hysteresis effects, viscoelasticity, and wear out effects of the mechanically exposed material.

This paper investigates the learning of reaching skills on such systems, i.e., to move the end-effector of the robot toward some desired position by changing the robot's posture. Successful control of such tasks can be well understood with the notion of internal models [5]. Once internal models are established for a certain task, a forward model predicts the consequence of a motor command, while an inverse model suggests a motor command necessary to achieve a desired outcome. Learning internal models from scratch requires exploration. In artificial systems, exploration is traditionally addressed by motor babbling [6]–[8]; that is, motor commands are randomly selected and their consequences are observed. This kind of exploration becomes very inefficient with increasing dimension of the sensorimotor space. The exploration can be significantly improved by active learning schemes [9], [10]. Although the risk of generating uninformative examples can be reduced with these methods, they assume that the sensorimotor space can be entirely explored. However, high-dimensional motor systems cannot be entirely explored in a lifetime.

How can a motor system, that cannot even be fully explored once, be mastered if it is non-stationary and changes constantly? Humans face that problem in their early sensorimotor development. While infants bootstrap their repertoire of sensorimotor skills, their bodies undergo massive changes in overall size, weight, segment lengths, and mass distribution. Therefore, investigations of infants' exploratory behavior can provide inspiration for algorithms to deal with high-dimensional and nonstationary control problems: It has been shown that infants explore by far not randomly or exhaustively as supposed by motor babbling. Rather, they attempt goal-directed actions already days after birth [11], even though they fail, which indicates a strong role of learning by doing. Infants learn to reach by trying to reach, which we refer to as goal babbling [12]. This strategy is highly beneficial for high-dimensional motor systems that are also highly redundant: tasks in sensorimotor learning are typically much lower dimensional than the motor systems themselves. Reaching can be done in an infinite number of ways, because human bodies as well as modern robotic systems have more degrees of freedom than necessary to solve the task. While this redundancy is often considered a problem for sensorimotor learning [13], [14], it also reduces the demand for exploration. If there are multiple ways to achieve the same result there is no inherent need to know all of them. Goal babbling allows to focus on behaviorally relevant data and leave out redundant choices as long as there are known solutions that work.

A. Related Work

The general idea to mimic infants' efficient sensorimotor learning is to likewise perform goal-directed exploration from

Manuscript received May 15, 2012; revised August 28, 2013; accepted October 20, 2013. Date of publication November 20, 2013; date of current version May 15, 2014.

The authors are with the Research Institute for Cognition and Robotics, Bielefeld University, Bielefeld 33615, Germany (e-mail: fmrolf@cor-lab.uni-bielefeld.de; jsteilg@cor-lab.uni-bielefeld.de).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2013.2287890

the very beginning, i.e., to perform goal babbling. Instead of performing random movements for the sake of learning, the robot chooses actions by trying to achieve goals, and performs ongoing learning. Goal-directed exploration has been a part of many learning schemes, but only been possible with prior knowledge [15], [16] or nongoal-directed pretraining [7], [17]. Only recently, models have been proposed that investigate a consistent, goal-directed bootstrapping of internal models for sensorimotor control. In [18], an associative memory related to forward models is learned. A full forward model, as well as a full feedback model can, however, only be learned with exhaustive exploration. The most direct way to perform partial exploration of the sensorimotor space is to start with exactly one solution that is learned directly in an inverse function. In [12], we have introduced a model for learning such functions with goal babbling based on batch-gradient learning. Yet, online learning is particularly beneficial in this scenario: since learning instantaneously results in more informative samples during goal-directed exploration, online-learning constitutes a positive feedback loop, which allows for enormous speedups [19]. The method has so far demonstrated the fastest bootstrapping performance among the proposed algorithms: it can bootstrap the control of 2-D control tasks on high-dimensional systems (e.g., 50 degrees of freedom) within a few hundred exploratory movements, which is competitive with human learning performance [20]. This performance stands in contrast to thousands to hundred thousands of movements necessary for other algorithms [12], [18], [21]–[23] when faced with comparable tasks. The general advantage of goal-directed over random exploration has been confirmed in several other studies for the learning of forward models [21], [22] as well as feedback control models [23]. It describes an incremental and ongoing process that supersedes any decision when to perform a relearning or to perform a distinct exploration phase. Hence, it provides the basis for an efficient mastery of high-dimensional nonstationary motor systems.

Yet, so far there has been no quantitative evidence for the success of these methods in practical real-world scenarios. Most of the studies investigated pure simulations with an ideal execution of actions without noise or delays, stationary system behavior, and comfortable ranges. In [24], it was shown that the approach to learn inverse models can deal with nonstationary behavior in an otherwise simple simulation task if learning occurs from small batches of data, which still requires too much data to be practically feasible. The only physical robot experiments have been shown in [23] and [25], but only with qualitative results. Both studies did: 1) not provide an actual assessment of the control accuracy after learning (except for a single example trajectory with only moderate accuracy in [23]) and 2) no details about the development of this accuracy over the course of learning.

Simultaneously, no control concept for the bionic handling assistant (BHA) (BHA, see Fig. 1) has been introduced so far. This practically relevant scenario comprises several properties, such as elastic motions and narrow and changing actuation ranges, that are very hard to model analytically. The central contributions of this paper are: 1) to show the success of goal babbling on this challenging platform, which provides the first quantitative and detailed results of such schemes in a real-world scenario and 2) to introduce the first functioning control concept for this practical robot platform, which also includes a novel integration of such learning with feedback control mechanisms.

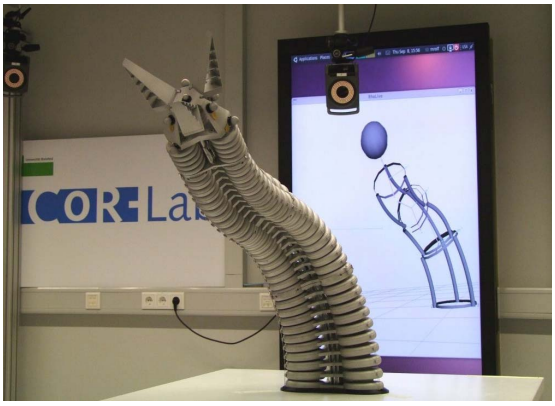


Fig. 1. BHA mimics an elephant trunk.

B. Overview

We introduce the robot platform and its learning problem in Section II. The nonstationary behavior is shown to express mostly in terms of the actuation ranges: the limits in which each actuator can be moved change over time due to the viscoelasticity of the robot's material. The exploration and learning approach based on [19] is described in Section III. The main contribution is described in Section IV: we show how the learning performs on the physical, nonstationary robot and present an in-depth analysis of the particular challenges on this exemplary, but for bionic robots prototypical platform. We discuss how the interplay of generalization during learning and unknown actuation ranges can cause execution failures on the highly constrained actuators—and how learning nevertheless finds appropriate solutions. We introduce a novel combination of learned models and feedback control, due to which the BHAs end-effector can be controlled with only a few millimeters error after learning. While the precise nonstationarities on the real robot can hardly be determined during its operation, Section V further investigates the impact of nonstationary behaviors in simulation experiments. We investigate the degeneration of actuation ranges, sensory-drifts, and simulated morphological growth. Section VI concludes this paper with a discussion.

II. BHA Setup

The robotic platform used in this paper is the BHA [3] which is a new, award-winning [26] continuum robot platform inspired by elephant trunks and manufactured by Festo (see Fig. 1). The robot is pneumatically actuated and made almost completely out of polyamide, which makes it very flexible and lightweight (ca. 1.8 kg).

A. Actuation and Sensing

The robot comprises three main segments, each with three pneumatic bellow actuators, a ball joint as wrist, also actuated

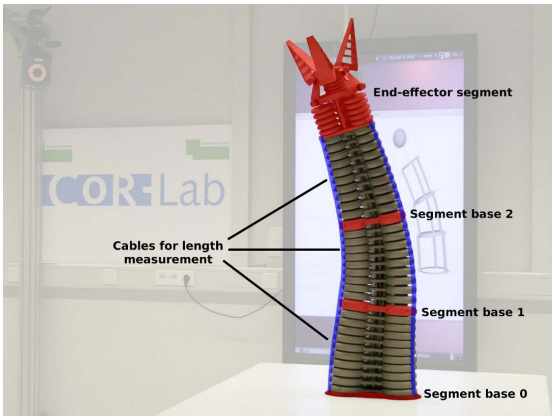


Fig. 2. Kinematic structure of the BHA comprises three main segments, each consisting of three parallel pneumatic bellow actuators. The length of these actuators can be determined with cable potentiometers.

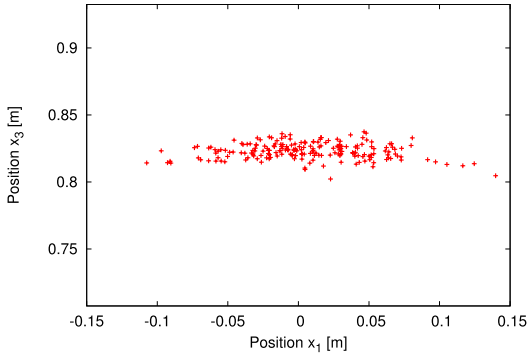


Fig. 3. Effector positions for an i.i.d. variation of the nine actuator lengths. Already a deviation of 5 mm on each actuator-length causes several centimeters sideward movement of the end-effector, but only small stretching movements.

by three actuators, and a three finger gripper actuated by one bellow actuator. This paper only uses the main segments, so that we use $m = 9$ actuated degrees of freedom. Each actuator can be supplied compressed air, which unfolds and extends the actuator. The combination of three actuators per segment then allows to bend, and—in contrast to standard robots with revolute joints–stretch the entire robot.

For a reliable positioning, it is not sufficient to control the pressure alone: friction, hysteresis, and nonstationarities can cause largely different postures when supplying the same pressure several times. In particular, during dynamic movements the pressure is not sufficient to determine the posture or position of the robot, since it only expresses a force on the actuators. This force reaches an equilibrium with the mechanical tension of the bellows after some time, so that the robot stands still. This physical process can, however, take up to 20 s because of a strong mechanical interplay between different actuators. Since pressure does not provide reliable information about the robot's position and movement in space, we are solely concerned with the geometric information from the BHAs length-sensors (see Fig. 2). Each one cable-potentiometer spans the range from the trunk's base to the top end of each of the nine actuators. Hence, they allow to measure the outer length of the trunk along the actuators between the segment base 0 and 1, between 0 and 2, and between 0 and the end-effector segment. For control and learning we do not use these measurements directly, but subtract the lengths within each bundle of potentiometers from each other to get the outer length of each individual bellow actuator (e.g., length 1–0 from length 2–0 to get the length 2–1 of the middle actuator). Although they do not allow a direct actuation-like pulling, the length values can be controlled by adjusting the pressure in each actuator. Our system comprises a length-controller that performs this task automatically and itself was learned beforehand [27]. For the nine length values in the main segments, we generally refer to the desired length as $q^* \in \mathbb{R}^9$. The actually measured length is referred to as $q \in \mathbb{R}^9$, which might differ from q^* because of sensory noise, or q^* being not yet reached or being not reachable at all.

The forward kinematics function of this robot is not exactly known analytically, although approximations exists (see Section V). For the kinematic control of the robot, we consider the 3-D position of the end-effector. Using the position only (without orientation) is largely sufficient to grab objects with the widely opened elastic fingers of the BHAs gripper, and already requires to fully exhaust the narrow limits of the bellow actuators (see Section II-C). For our experiments, we measure the end-effector position with a VICON motion tracking system [28]. Auto-reflective markers allow to measure the position with high accuracy using triangulation. We use the central position inside the gripper's palm as point of reference and refer to its cartesian value as $x \in \mathbb{R}^n, n = 3$. We denote the robot's implicit forward function as $f(q) = x$. This function cannot be evaluated directly, but examples x and q can be observed on the physical robot.

B. Achievable Accuracy

Although the length of the actuators can be controlled, there are limitations to the positioning accuracy that need to be considered for learning experiments. The first important property of the BHAs morphology is that even minimal changes of the actuator lengths can lead to large, and direction-wise inhomogeneous changes of the effector position. To illustrate this phenomenon, we recorded the end-effector position for 200 random postures, each drawn independent identically distributed (i.i.d.) from normal distribution around a stretched position $q_i = 0.225$ m $\forall\ i = 1 \ldots 9$ with standard-deviation 5-mm per actuator. Fig. 3 shows the resulting positions of the end-effector from a sidewise perspective. The resulting distribution extends to almost 15-cm sideward deviation (the first axis, x_1), while top/down stretching movements (x_3) only vary within ± 2 cm. The standard deviations of the generated distribution are 4.4 cm in x_1 and x_2 direction and 0.6 cm in x_3 direction. The large amplitude of sideward movements implies a very high sensitivity of the end-effector position to length-changes. In reverse, a positioning of the end-effector with low deviation (e.g., 1 cm) requires a control of the actuator lengths with submillimeter accuracy. This is clearly difficult to achieve on the BHA due to long delays in the pneumatic actuation,

and strong sensory noise in the length-sensing (ca. 1-mm amplitude).

To obtain a baseline for positioning accuracy of the BHA, we chose $P = 20$ entirely random postures q_p. These postures were set as target for the length-controller [27]. Due to the slow steady-state dynamics of the physical deformation process, it is practically impossible to determine whether the robot's deformation due to an applied pressure has already converged. Therefore, we chose to apply each posture as target to the length-controller for entire 20 s, which is in our experience long enough to reach a physical equilibrium. This procedure was repeated $R = 20$ times with different permutations of p. Each time we recorded the resulting Cartesian end-effector position x_p^r. We evaluated the distance of these positions from the average position per q_p

$$\overline{x}_p = \frac{1}{R}\sum_r x_p^r$$
$$D = \frac{1}{P}\sum_p \frac{1}{R}\sum_r ||x_p^r - \overline{x}_p||$$

where $||\cdot||$ is the Euclidean norm. Results show that $D = 0.0047$ m. Hence, the end-effector can only positioned with approximately 5-mm accuracy.

C. Nonstationary Actuation Ranges

A central problem for the control of the BHA is that the limits, in which the actuator lengths can be controlled, are not known and change over time. Limits for the lowest-level physical actuation, i.e., the pressure, are easily formulated: since the actuation only works with over-pressure, each actuator has a minimum pressure of 0 bar. The maximal admissible pressures that allow for a safe operation (i.e., that do not burst the bellows) are 0.9, 1, and 1.2 bar for the first, second, and third segments. These maximum pressures are chosen by the manufacturer and cannot be exceeded. Hence, the set of possible pressure combinations is a hyper-rectangle in nine dimensions. In contrast, the set of possible length combination is clearly not a hyper-rectangle since each length is the result of a strongly nonlinear physical deformation process. This is shown in the first part of Table I: combinations of minimum/maximum pressure were supplied to the three actuators in the third segment, and the resulting three actuator lengths were recorded. Two effects are clearly visible.

1) The different actuators have different limits, even within the same segment, due to viscoelasticity and wear-out effects. This is particularly visible in the last line of the table, where maximum pressure for each actuator generates significantly different lengths.
2) There are significant interdependencies between the limits of different actuators: the maximum reachable length (i.e., the length for maximum pressure) depends on the length of the other actuators.

Such combinations of minimum and maximum pressure give some insight into the structure of the length ranges. Yet, the analytic shape of the set of possible length combinations is not known. We refer to this set as $\mathbf{Q} \subset \mathbb{R}^9$. Each vector in $\mathbf{Q}$ represents a length-combination that is reachable for the robot. Each vector that is not in $\mathbf{Q}$ cannot be reached.

TABLE I

MEASURED ACTUATION LIMITS FOR SEGMENT 3 BEFORE AND AFTER THE LEARNING EXPERIMENTS. CHANGES OF MORE THAN 2.5 mm ARE MARKED WITH *, CHANGES OF MORE THAN 5 mm WITH **

Before the experiments

Pressure [bar]			Length $[m]$		
0	0	0	0.1825	0.1873	0.1834
0	0	**1.2**	0.1727	0.1782	**0.2513**
0	**1.2**	0	0.1748	**0.2681**	0.1749
1.2	0	0	**0.2545**	0.1757	0.1760
			.....		
1.2	**1.2**	**1.2**	**0.2476**	**0.2647**	**0.2338**

After the experiments

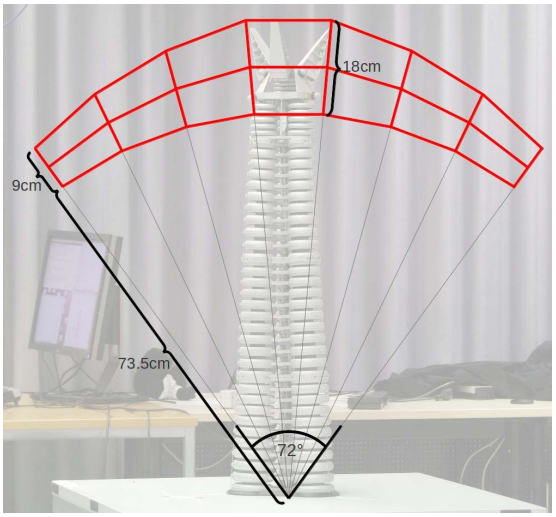
Pressure [bar]			Length $[m]$		
0	0	0	0.1839	0.1870	0.1859*
0	0	**1.2**	0.1750	0.1783	**0.2581****
0	**1.2**	0	0.1754	**0.2709***	0.1744
1.2	0	0	**0.2615****	0.1761	0.1771
			.....		
1.2	**1.2**	**1.2**	**0.2538****	**0.2654**	**0.2388****

Q is not only not known, it is not stationary. The upper part of Table I was recorded before the experiments described in Section IV. We repeated the same procedure after the experiments and found that the limits have changed significantly (lower part of Table I). For instance, the maximum values in the last column have changed by 6–7 mm, which is substantially above sensory noise and can cause large changes of the effector positions (see Fig. 3). This change is caused by the viscoelasticity of the bellows' polyamide material: it is not perfectly elastic, but has a certain memory of its recent form. This corresponds to a spring with changing force constant, so that the same pressure (e.g., force on the spring) can cause different elongations over the course of time. For practical experimentation with the BHA means that whenever some posture q^* is desired, it is not even clear whether the posture can be reached.

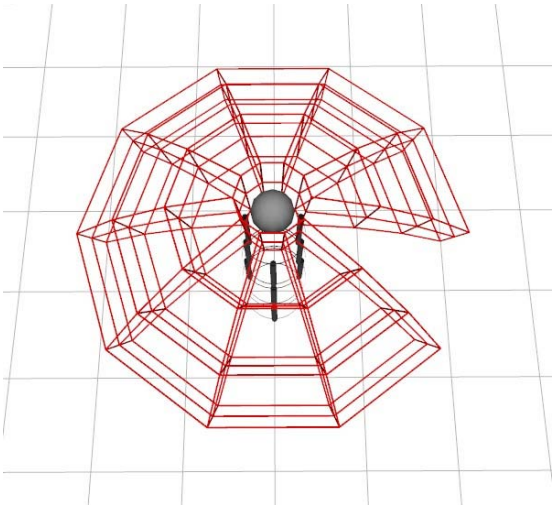
D. Kinematic Learning Problem

Reaching for some desired Cartesian position $x^* \in \mathbb{R}^3$ with this robot means to find some posture, i.e., a combination of lengths q, that results in an end-effector position $x = x^*$. In the following experiments, we consider the learning of reaching skills for the volume $\mathbf{X}^*$ of targets shown in Fig. 4. To describe this volume using a finite set of representative targets, we chose a grid with $K = 120$ vertices. A side view is shown in Fig. 4(a): the representatives are the 24 vertices of the red grid, which is shown in relation to the BHA. A 3-D workspace description is constructed from this plain grid by rotating it around five different angles. Fig. 4(b) shows the resulting grid in a 3-D view from above. Note that the gaps in the 3-D visualization are only for visual orientation.

The goal of the experiments is to learn an inverse model $g(x^*)$ for the volume $\mathbf{X}^*$ enclosed in this grid. During exploration, target positions x^* are interpolated between the K grid vertices. Furthermore, exploratory noise distributes observations also around such paths to eventually cover the



(a)



(b)

Fig. 4. Inverse model is learned for the volume of targets $\mathbf{X}^*$ enclosed in the red grid, shown (a) from a sidewise perspective in 2-D and (b) from a top view in 3-D.

entire volume. The inverse model is asked to estimate a posture q^* that allows to move the effector to x^*: $f(g(x^*)) = x^*$. During the process of learning, the resulting position $f(g(x^*))$ will generally differ from x^*. Hence, the central measure of learning progress is the Cartesian performance error, which measures the distance between the actual and the desired positions at the K representatives

$$E^X = \frac{1}{K} \sum_{k=0}^{k<K} ||f(g(x_k^*)) - x_k^*||. \tag{1}$$

While all evaluations are performed in Cartesian coordinates in order to provide easily understandable distances in meters, the learning is performed in a different coordinate system. Since the exploration is based on the sampling of continuous paths (see the following section) it is desirable to have a convex workspace, which allows to sample a linear path between any two points. To achieve that for the given workspace, we change the representation to an angular coordinate system. The following transformation is applied before spatial coordinates $x = (x_1, x_2, x_3)^T$ are used for learning

$$\psi(x) = (\ \mathrm{sgn}(x_3) \cdot ||x||,\ \sphericalangle(x, \mathbf{u_1}),\ \sphericalangle(x, \mathbf{u_2})\ )^T$$

where $\mathbf{u_1}$ and $\mathbf{u_2}$ are the unit vectors along the first and second axis. The first component of ψ: $\mathbb{R}^3 \to \mathbb{R}^3$ represents the radial component, i.e., the distance of some point from the BHAs base. The last two components express angles. The workspace $\psi(\mathbf{X}^*)$ after the transform is a convex set so that linear paths can be sampled without leaving the set.

III. Online Goal Babbling

The general idea for the coordination of the BHA is to learn an inverse model $g : \psi(\mathbf{X}^*) \to \mathbf{Q}$ that suggests motor commands q^* for reaching targets $\psi(x^*)$. Such a function g can be learned by exploring actions, observing their outcomes, and then using the collected data for a supervised learning step. During exploration, these examples are generated iteratively over time steps t. We denote the motor commands that are sent to the robot during exploration as q_t^*. The resulting postures q_t and effector positions x_t are measured for learning, so that that q_t and x_t correspond to an evaluation of the robot's implicit forward function

$$x_t = f(q_t). \tag{2}$$

Examples (x_t, q_t) can then be used for supervised adaption of the inverse estimate $g(\psi(x^*), \theta)$, where θ is a set of parameters adaptable by learning. Since learning in this paper is consistently done in the angular coordinates as described above, we write $\psi(x_t) = \psi_t$ and $\psi(x_t^*) = \psi_t^*$ as short notation.

Goal babbling refers to the way these examples are generated, i.e., how actions q_t^* are chosen using goal-directed movement attempts. Section III-A introduces the basic formalism to perform such goal-directed movements. Section III-B explains how this formalism is used to organize continuous movement paths in time t, and Section III-C describes how exploratory noise is injected into this mechanism. Finally, Section III-D details how the inverse model is represented and learned based on the generated training data. The entire algorithm is based on [19] with some minor modifications to improve the efficiency in challenging physical robot setups like the BHAs.

A. Goal-Directed Movements

To generate examples, goal babbling starts with an initial inverse estimate $g(\psi^*, \theta_0)$ that always suggests some comfortable home posture: $g(\psi^*, \theta_0) = \mathrm{const} = q^{\mathrm{home}}$. Then, continuous paths of target positions ψ_t^* through $\psi(\mathbf{X}^*)$ are iteratively chosen by interpolating between the K representative points. The system then tries to reach for these targets, which corresponds to infants' early goal-directed movement attempts. For that purpose, the inverse estimate is evaluated as expressed in the fundamental equation of goal-directed exploration

$$q_t^* = g(\psi_t^*, \theta_t) + E_t(\psi_t^*). \tag{3}$$

The command q_t^* is sent to the length controller, the outcomes q_t and ψ_t are observed, and the parameters θ_t of the inverse estimate are updated based on the example (ψ_t, q_t) immediately before the next example is generated. In contrast to earlier simulation studies [19] it is crucial to make the distinction between q_t^* and q_t at this point: the command q_t^* might not be executable, or might not yet be reached at the time of measurement. Hence, only (ψ_t, q_t) but not (ψ_t, q_t^*) represents a sample of the ground truth forward function that is useful for learning. The perturbation term $E_t(\psi^*)$ adds exploratory noise in order to discover new positions or more efficient ways to reach for the targets. This allows to unfold the inverse estimate from the home posture and finally find correct solutions for all positions in the volume of targets $\psi(\mathbf{X}^*)$.

B. Path Generation

A major aspect of goal babbling is how to choose target positions. We do so by generating continuous, piecewise linear target movements through $\psi(\mathbf{X}^*)$. The initial target ($t = 0$) is the effector position corresponding to the home posture: $\psi_0^* = \psi(f(q^{\text{home}}))$. In the first movement, the system tries to move along a path toward another target Ψ_1^* which is randomly chosen from the K representative points of $\psi(\mathbf{X}^*)$. This path is generated by interpolating linearly between ψ_0^* and Ψ_1^*. Afterward, a new target Ψ_2^* is chosen from $\psi(\mathbf{X}^*)$ and the second movement is attempted between Ψ_1^* and Ψ_2^*. This movement is generated with a fixed difference δ_ψ between the successive samples: as long the next endpoint Ψ_l^* is more than δ_ψ away from the last target ψ_t^*, it receives an update

$$\psi_{t+1}^* = \psi_t^* + \frac{\delta_\psi}{||\Psi_l^* - \psi_t^*||} \cdot (\Psi_l^* - \psi_t^*). \tag{4}$$

When ψ_t^* was closer than δ_ψ to Ψ_l^*, we set $\psi_{t+1}^* = \Psi_l^*$, and a new Ψ_{l+1}^* is chosen to continue. An example is generated for each of these targets according to (2) and (3). Earlier simulation work [19] used an interpolation between the endpoints with a fixed number of intermediate samples. The sampling with fixed step-length (varying number of intermediate samples) proposed here generates a more uniform distribution and keeps a movement speed that is neither too fast to be executable, nor too slow, preventing a bad signal-to-noise ratio.

In nonlinear redundant domains, it is generally possible to generate inconsistent examples with same effector pose but different joint angles. Learning from such examples leads to invalid solutions [16]. We have previously shown in [12] that the structure of goal-directed exploration allows to resolve such inconsistencies using a weighting scheme

$$w_t^{\text{dir}} = \frac{1}{2}\left(1 + \cos \sphericalangle(\psi_t^* - \psi_{t-1}^*,\ \psi_t - \psi_{t-1})\right) \tag{5}$$

$$w_t^{\text{eff}} = ||\psi_t - \psi_{t-1}|| \cdot ||q_t - q_{t-1}||^{-1} \tag{6}$$

$$w_t = w_t^{\text{dir}} \cdot w_t^{\text{eff}}. \tag{7}$$

w_t^{dir} measures whether the actually observed movement and the intended movement have the same direction. w_t^{eff} measures the kinematic efficiency of the movement and assigns high weight to examples that achieve a maximum of effector movement with a minimum of joint movement. For learning, each example (ψ_t, q_t) is weighted by w_t. In addition to resolving inconsistencies, the weighting guides the redundancy resolution in redundant domains: efficient movements will dominate the learning in the long term and cause the inverse estimate to select smooth and comfortable solutions [12].

A special kind of movement is used to prevent drifts into irrelevant regions of the sensorimotor space. Similar to the infants practicing their motor skills, the system returns to a stable point after a while and starts to practice again. With a probability p^{home}, the next movement after a target Ψ_l^* has been applied is not another goal-directed movement. Instead, the system returns to its home posture. This kind of movement leads to a repetitive presentation of examples close to the home posture and forces the inverse estimate to reproduce these postures for goal-directed movements. It acts as a developmentally plausible stabilizer that helps to stay in known areas of the sensorimotor space [12], [18]. We model this movement as a linear path in the space of postures $\mathbf{Q}$ to get smooth and continuous behavior for online learning: the system moves from the last actuated posture q_t^* to its home posture q^{home}, whereas (3) is replaced by the following:

$$q_{t+1}^* = q_t^* + \frac{\delta_q}{||q^{\text{home}} - q_t^*||} \cdot (q^{\text{home}} - q_t^*) \tag{8}$$

if q_t^* is not closer than δ_q to the home posture, and $q_{t+1}^* = q^{\text{home}}$ if it is close enough. For every generated motor command q_t^*, the resulting posture and effector position is observed (2) and learning is applied online in the same way as for goal-directed movements. These examples are only weighted with w_t^{eff}, because targets ψ_t^* for the evaluation w_t^{dir} do not exist during this homeward movement. After the home posture has been reached, a goal-directed movement is attempted from the initial target $\psi_{t+1}^* = \psi(f(q^{\text{home}}))$.

C. Structured Continuous Variation

To find kinematic solutions for all target positions, it is necessary to consider exploratory noise, or rather perturbations of the motor system [12], [29]. Such perturbations arise naturally in physical systems and lead to the exploration of new postures that would not be suggested by the inverse estimate. Physical perturbations typically lead to smooth variations of the intended movements. At any point in time, we model this effect by adding a small, randomly chosen linear function to the inverse estimate

$$E_t(\psi^*) = A_t \cdot \psi^* + b_t, \quad A_t \in \mathbb{R}^{m \times n},\ b_t \in \mathbb{R}^m. \tag{9}$$

Initially, all entries e_0^i of the matrix A_0 are chosen i.i.d. from a normal distribution with zero mean and variance σ^2. To explore the local surrounding of the inverse estimate, we vary these parameters slowly with a normalized Gaussian random walk. A small value δ_{t+1}^i is chosen from a normal distribution $N(0, \sigma_\Delta^2)$ with $\sigma_\Delta^2 \ll \sigma^2$, and added to the previous value e_t^i. The variance of the resulting value is the sum of the individual variances $\sigma^2 + \sigma_\Delta^2$. We normalize with the factor

$\sqrt{\sigma^2/(\sigma^2+\sigma_\Delta^2)}$ to keep the overall deviation stable at σ

$$e_0^i \sim N(0,\sigma^2), \quad \delta_{t+1}^i \sim N(0,\sigma_\Delta^2)$$
$$e_{t+1}^i = \sqrt{\frac{\sigma^2}{\sigma^2+\sigma_\Delta^2}} \cdot (e_t^i + \delta_{t+1}^i) \sim N(0,\sigma^2).$$

The same process generates vectors b_t with deviations $\sigma^{(b)}$ and $\sigma_\Delta^{(b)}$.[1] Hence, $E_t(\psi^*)$ is a slowly changing linear function. It is smooth at any time, which is important for the evaluation of the weighting scheme (5) and (6). It is furthermore zero centered and limited to a fixed variance, which leads to a local exploration around the inverse estimate.

D. Incremental Regression Model

For learning, a regression mechanism is needed to represent and adapt the inverse estimate $g(\psi^*)$. The goal-directed exploration itself does not require a particular functional form or other details of this regressor, such that in principal any regression algorithm can be used. For a safe and incremental online learning, we have chosen a local-linear map [30] for our experiments. The inverse estimate consists of different linear functions $g^{(k)}(\psi)$, which are centered around prototype vectors $p^{(k)}$ and active only in its close vicinity, which is defined by a radius d. The function $g(\psi^*)$ is a linear combination of these local linear functions, weighted by a Gaussian responsibility function $b(\psi)$

$$g(\psi^*) = \frac{1}{n(\psi^*)} \sum_{k=1}^{K} b\left(\frac{\psi^* - p^{(k)}}{d}\right) \cdot g^{(k)}\left(\frac{\psi^* - p^{(k)}}{d}\right)$$
$$b(\psi) = \exp\left(-||\psi||^2\right), \quad n(\psi^*) = \sum_{k=1}^{K} b\left(\frac{\psi^* - p^{(k)}}{d}\right)$$
$$g^{(k)}(\psi) = W^{(k)} \cdot x + o^{(k)}.$$

The normalization $n(\psi^*)$ scales the sum of influences of the components to unity, which is known as soft-max.

The inverse estimate is initialized with a single local function with center $p^{(1)} = \psi(f(q^{\text{home}}))$ that outputs the constant value q^{home} ($W^{(1)} = 0_{m\times n}$, $o^{(1)} = q^{\text{home}}$). New local functions and prototypes are added dynamically. Whenever the learner receives an input ψ^*, that has a distance of at least d to all existing prototypes, a new prototype $p^{K+1} = \psi^*$ is created. To avoid abrupt changes in the inverse estimate, the function $g^{K+1}(\psi)$ is initialized such that its insertion does not change the local behavior of $g(\psi^*)$ at the position ψ^*. We set the offset vector o^{K+1} to the value of the inverse estimate before the insertion of the new local function: $o^{K+1} = g(\psi^*)$. The weight matrix is initialized with the Jacobian matrix $J(\psi^*) = \partial g(\psi^*)/\partial \psi^*$ of inverse estimate: $W^{K+1} = J(\psi^*)$.

In each time step, the inverse estimate is fitted to the current example (ψ_t, q_t) by reducing the weighted square error

$$E_w^Q = w_t \cdot ||q_t - g(\psi_t)||^2.$$

[1]Earlier work used identical amplitudes for A_t and b_t. We split these values for the BHA to account for different numerical amplitudes of the goals, with which A_t is multiplied. This keeps the same ratio between $A_t \cdot \psi^*$ and b_t that was found useful in [19].

TABLE II
PARAMETERS USED FOR EXPLORATION AND LEARNING

Target step width	δ_ψ	0.01m
Posture step width	δ_q	0.002m
"Go-Home" probability	p^{home}	0.1
Perturbation amplitude	σ	0.0025
Perturbation amplitude	$\sigma^{(b)}$	0.005
Perturbation change-rate	σ_Δ	$0.1 \cdot \sigma$
Perturbation change-rate	$\sigma_\Delta^{(b)}$	$0.1 \cdot \sigma^{(b)}$
Local learning distance	d	0.1
Learning rate	η	0.05

The parameters $\theta = \{W^{(k)}, o^{(k)}\}_k$ of $g(\psi^*)$ are updated using online gradient descent on E_w^Q with a learning rate η

$$W_{t+1}^{(k)} = W_t^{(k)} - \eta \frac{\partial E_w^Q}{\partial W^{(k)}} \qquad o_{t+1}^{(k)} = o_t^{(k)} - \eta \frac{\partial E_w^Q}{\partial o^{(k)}}.$$

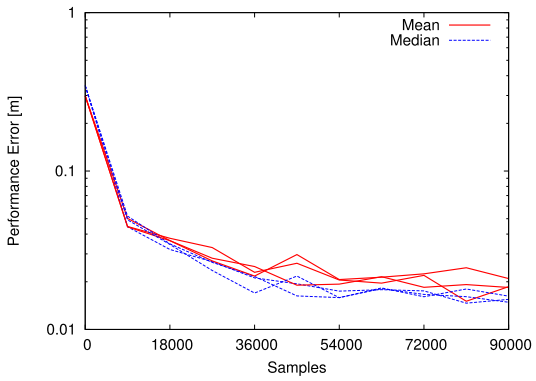
IV. REAL-WORLD BHA EXPERIMENTS

This section presents experiments with online Goal Babbling on the physical BHA robot. During the experiments, the robot underwent a change of actuation ranges, as detailed in Section II-C so that learning operated on a nonstationary system. We first illustrate how the reaching performance develops during learning. We then present a method for local error correction, which reduces the residual errors due to nonreachable target postures.

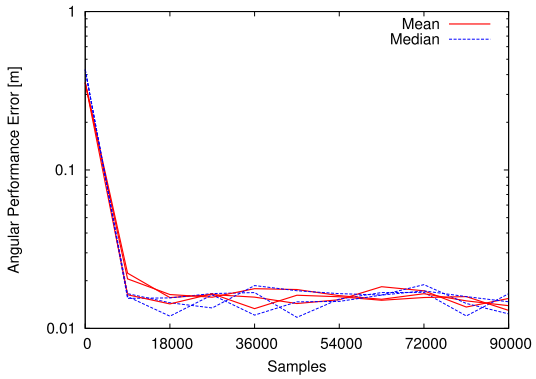
A. Learning on the Nonstationary Robot

We applied the exploration and learning algorithm on the BHA in three independent trials. The workspace description $\mathbf{X}^*$ was used, as illustrated in Section II-D. All parameter values are shown in Table II. The most influential parameters are the amplitude of the perturbation terms, which has been extensively investigated in [12], and learning rate, which has been focus of investigation in [19]. The sampling rate on the robot is 5 Hz: in each second, five targets ψ_t^* are generated and the resulting samples are used for learning. With the target step length $\delta_\psi = 0.01$ m this corresponds to a target velocity of 5 cm/s, which is suitable for the robot. In each trial, the method used $T = 90\,000$ samples, which corresponds to five hours real time.

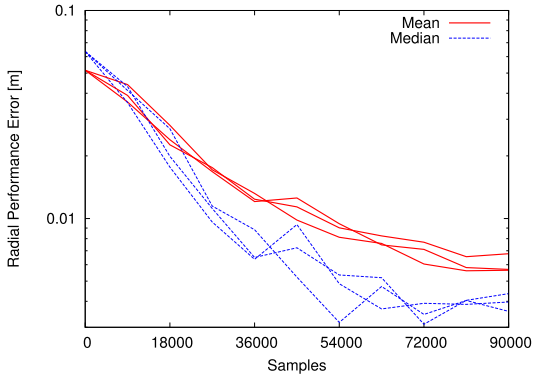
Every 9000 samples the learning was interrupted to measure the current performance on the $K = 120$ targets in Fig. 4. The current inverse estimate $g(\cdot, \theta_t)$ was used to estimate the posture $q_k^* = g(\psi_k^*, \theta_t)$. The length controller had 20 s time to reach and stabilize q_k^*. Statistics of the Cartesian performance errors between the targets x_k^* and the actually observed positions x_k are shown in Fig. 5(a) for all three trials. The initial error is approximately 30 cm, which corresponds to the average distance of the home position, in which the learner is initialized, and the different target positions. Subsequently, the exploration procedure reduces the error rapidly. After $T = 90\,000$, the errors consistently reach a mean level of ca. 2 cm and a median level of ca. 1.5 cm in all three trials. For an average robot-length of 80 cm this corresponds to 2%–3% relative error, which already includes the general



(a)



(b)



(c)

Fig. 5. (a) Cartesian performance error is reliably reduced in all three trials. The mean reaches approximately 2 cm and the median value 1.5 cm. A decomposition into (b) angular and (c) radial components shows that the 2-D angular subproblem is solved already within the first 9000 samples.

execution uncertainty of 5 mm (see Section II-B). The learning clearly succeeds to bootstrap the reaching skill on the robot. The remainder of this section closely investigates the details of this performance curve, the reasons for residual errors, and how they can be removed by further exploitation of the learned inverse model with a feedback controller.

Figs. 6 and 7 show a more detailed view on the first trial. Fig. 6 shows the observed effector positions x_t during the entire learning procedure. Starting from the home posture and the discrete set of representative targets, the exploration and learning procedure eventually generates movement paths through the entire volumetric workspace. Thereby both the

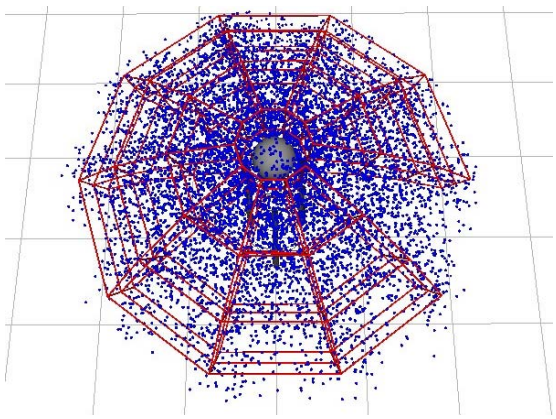


Fig. 6. Examples $x_t = \psi^{-1}(\psi_t)$ generated during goal babbling in the first trial. Only every 10th sample is shown, i.e., one example for every two seconds of exploratory time. While the 9-D space of motor commands cannot be exhaustively sampled, goal babbling achieves a comprehensive sampling of the 3-D workspace. Hence, a continuous inverse model for this space can be learned.

interpolation between the goals (4) as well as the addition of exploratory noise (3) contribute to the coverage of the volume. After the successive presentation of the 90 000 examples a continuous inverse model is learned for reaching within the volume. Histograms of the performance error are shown for $t = 0$, $t = 9000$, and $t = 90\,000$ in Fig. 7. The initial histogram simply shows the distances of the initial posture from the four rings of the target grid. Further histograms show that the error is reduced continuously, but also that few, isolated targets generally show an comparably high residual error. The right side of the figure shows the behavior of the learner in the 3-D space. The red grid again shows the set of targets. The blue grids show the measured behavior of the inverse estimate when trying to reach for the targets, i.e., the observed positions $x_k = f(g(\psi_k))$. Already after $t = 9000$ the positions are spread out along the angular directions, but do not yet cover the volume of the target set. After $t = 90\,000$ the learner has also discovered how to stretch along the radial axis: target and actual grid are in good correspondence.

Stretching seems to be a simple movement on the robot: in a straight position all actuators need to be extended and the effector moves upward. It is the most difficult movement: it requires a highly coordinated motor action, and the robot will deviate substantially if only one degree of freedom does not follow this movement. Due to the very restrictive actuation limits it is also necessary to include all three segments into the movement to reach from the very bottom of the workspace to the very top. In contrast, angular motions are much simpler and can be done in a lot of different ways. Due to the high sensitivity of the robot to movements in these directions (see Fig. 3) they are also easily discovered during autonomous exploration. Since the combination of goal-directed exploration and online learning unfolds a positive feedback-loop during the initial bootstrapping [19], the learner can basically master angular movements already after a few minutes. Radial stretching movements have lower sensitivity, which implies a lower gain in the feedback loop. Hence, it requires more time to learn this movement direction.

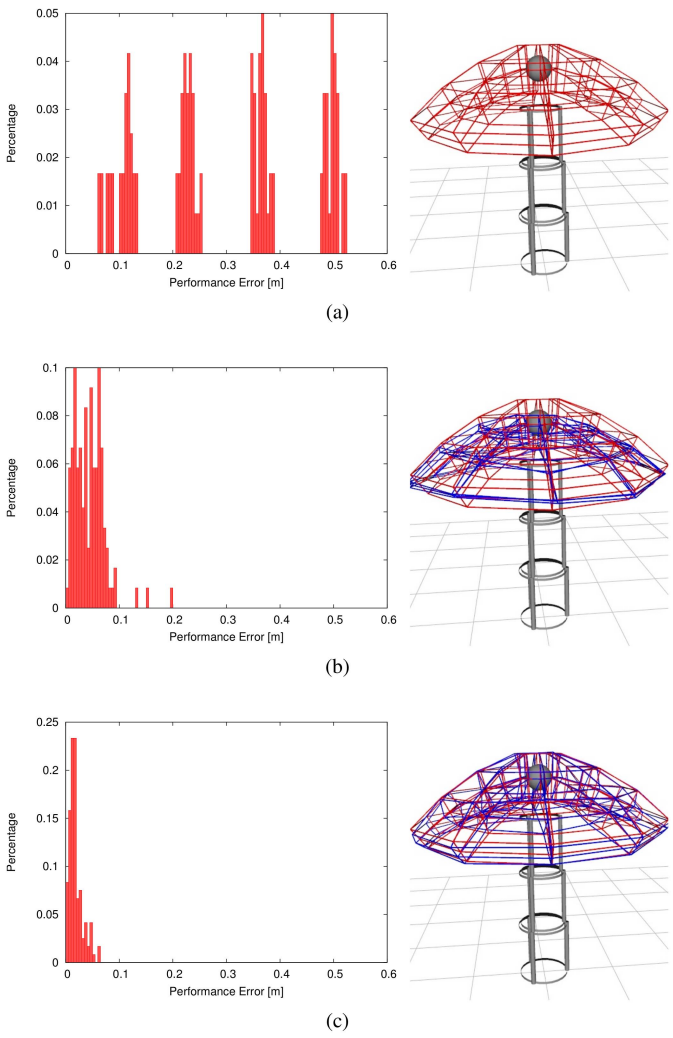


Fig. 7. Histograms of the (a) initial performance error, and the performance error (b) after $t = 9000$ and (c) after $t = 90\,000$ samples. While the initial histogram shows the ring structure of the target set, ongoing learning reduces the errors consistently. At $t = 9000$ the learner still has to make strong extrapolation, which lead to outliers (several points with errors above 10 cm), which are caused by execution failures $q \neq q^*$. Further learning consolidates these extrapolations.

This behavior occurs consistently over the three trials: Fig. 5(b) and (c) shows a decomposition of the performance error into angular and radial components. For the angular component, we projected both x_k and x_k^* on the unit-sphere with radius 1 m, such that the radial component is erased, and measured the Euclidean distance between the projected points. We evaluated this component only for the central of the three target layers (see Fig. 4). The top and bottom layers are not considered to blend out the difficulties of stretching movements for this evaluation. The radial error is the difference between the first components of $\psi(x_k)$ and $\psi(x_k^*)$, which is evaluated for all target positions. The plots show that the angular error component is reduced from 30 to 2 cm already in the first exploration episode, and further stabilizes around 1.3 cm. The bootstrapping and fine-tuning of radial movements takes significantly more time in all three trials. The difficulty to discover (and also control) stretching movements, while other directions are that simpler to find is very specific for the BHAs trunk morphology that combines bending and stretching. After all, this problem is solved by the exploration procedure.

TABLE III
CARTESIAN PERFORMANCE ERRORS WITHOUT AND WITH CARTESIAN FEEDBACK CONTROL ON TOP OF THE LEARNED INVERSE MODEL. THE CONTROLLER REMOVES ERRORS INDUCED BY EXECUTION FAILURES, AS INDICATED BY THE ERASED FAILURE CORRELATION

Feedforward Control with Learned Model

	Mean E^x $[m]$	Median E^x $[m]$	Failure-Corr.
Trial 1	0.0186	0.0155	0.832
Trial 2	0.0184	0.0149	0.728
Trial 3	0.0209	0.0162	0.845

Additional Feedback Control

	Mean E^x $[m]$	Median E^x $[m]$	Failure-Corr.
Trial 1	0.0074	0.0067	-0.045
Trial 2	0.0088	0.0080	0.101
Trial 3	0.0071	0.0064	0.017

B. Execution Failures

While the average performance during learning quickly reaches a good level, there remain rather isolated outliers. This behavior is particularly visible in Fig. 7(b), where a few targets are only reached with an error of more than 10 cm. These outliers are largely consolidated during learning, but a heavy-tail in the error-histogram remains [see Fig. 7(c)]. The reason for this behavior is grounded in the inevitable process of generalization and interference during the regression of g. During the initial bootstrapping of a motor skill this is an enormously useful mechanism: already based on the first examples x the learner generalizes and makes extrapolations for other targets x^*. These extrapolations are, of course, not perfect but allow a quick coverage of the workspace.

Once the learner has roughly covered the workspace, generalization can become more problematic due to the highly constrained actuation limits of the BHA. Moving through the entire set of targets requires to operate very closely to the limits of the possible length configurations $\mathbf{Q}$. Any data used for learning lies inside $\mathbf{Q}$ since the values of q_t have been observed on the robot. Interference, however, can cause a projection of g beyond $\mathbf{Q}$ for other positions x than that one currently used for learning (x_t). Suppose the current learning step is done on an example (x_t, q_t). Due to interference, the learner's output is changed at another position $x \neq x_t$ to $g(x) = q^*$ and $q^* \notin \mathbf{Q}$. When the inverse estimate is now used to reach for x, it would suggest q^*, which is not reachable. On the robot, this results in a different posture q. We refer to this mismatch $q^* \neq q$ as an execution failure. Due to the high angular movement sensitivity of the BHA already minor execution failures cause large deflections of the end-effector, and thus high Cartesian performance errors. The tight connection between the Cartesian errors and execution failures is shown in Table III. The upper part shows the final Cartesian errors for all three trials. The last column shows the failure correlation for the final evaluation after $t = 90\,000$

$$C_x^q = \varrho \left[\, ||x_k^* - x_k||, \; ||q_k^* - q_k|| \right]_k \tag{10}$$

where $\varrho \in [-1 : 1]$ is the Pearson correlation coefficient. It measures how well Cartesian errors are correlated with the

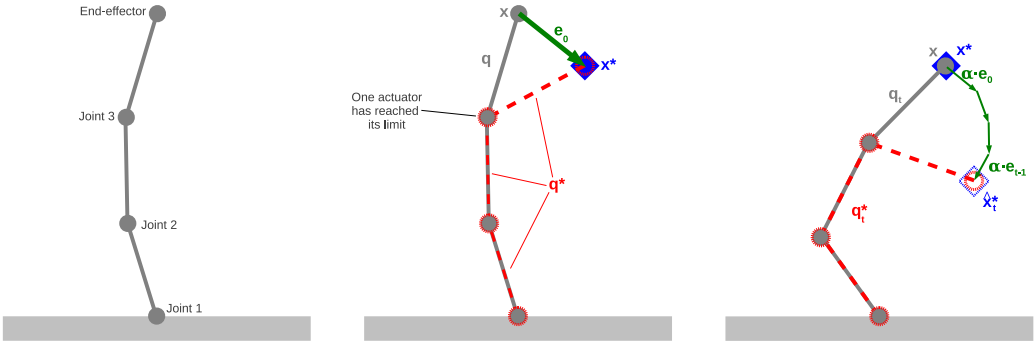


Fig. 8. (a) Cartesian feedback control on a simple robot with three revolute joints. (b) If an inverse model suggests a posture that cannot be executed due to actuation ranges. (c) Shifting of the target position allows to exploit the redundancy and nevertheless reach the target.

occurrence of execution failures. The table shows very high positive correlation in all three trials, which indicates that the largest Cartesian errors are indeed caused by execution failures.

C. Feedback Control for Local Error Correction

Although the interference is rather limited by the locally linear learning in our experiments, it is sufficient to cause the heavy-tailed error-distributions. Also, the projection outside $\mathbf{Q}$ is hardly avoidable, since $\mathbf{Q}$ is not even known and changes during operation. For our final experiment on the physical BHA, we propose a scheme that integrates the learned inverse models with an additional feedback controller, and demonstrate its ability to reduce residual errors caused by execution failures. Fig. 8(a) shows a simplified domain, with a planar arm comprising three revolute joints. An inverse model is used to reach a target position x^* [Fig. 8(b)]. The suggested posture q^* would indeed solve the task, but is not executable since the last joint has reached its actuation limit and cannot be bent further downward. The resulting posture q ends up in a position $x \neq x^*$. When an inverse model g has been established, feedback control can be applied in the Cartesian space without further learning: the target position is virtually shifted toward some value $\hat{x}_t^*$ and the posture $q_t^* = g(\hat{x}_t^*)$ is applied on the robot, which results in a posture q_t and an effector position x_t [see Fig. 8(c)]. The shifting of goals thereby follows the currently observed Cartesian error $e_t = x^* - x_t$, which is integrated over time

$$\hat{x}_0^* = x^*, \quad \hat{x}_t^* = \hat{x}_{t-1}^* + \alpha \cdot e_{t-1}.$$

This procedure is guaranteed to converge to the target position $\hat{x}_t^* = x^*$ if a shift of targets $\alpha \cdot e_{t-1}$ always results in an actual effector movement that has a positive angle to the desired movement ($\sphericalangle(e_{t-1}, x_t - x_{t-1}) < 90°$). If, however, the inverse estimate is not able to generate a positive movement direction, the control can diverge. It is possible that the limited actuator is driven even deeper into its limit during this feedback-controlled movement, since also the feedback-controller is not aware of $\mathbf{Q}$. One of the central strengths of our goal babbling algorithm is that it learns to efficiently distribute movements over all actuators [19]. This behavior can be exploited by

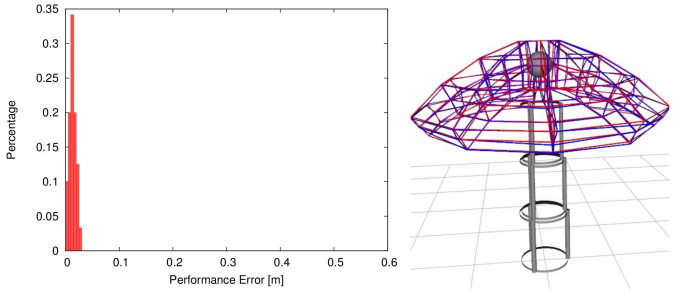


Fig. 9. Cartesian performance of a learned model when Cartesian feedback control is applied on top [compare Fig. 7(c)].

Cartesian feedback control, even if one actuator is blocked. As long as other actuators are still movable, the inverse estimate involves them to reach for $\hat{x}_t^*$, which brings the observed effector position x_t closer to x^* [see Fig. 8(c)].

We evaluated the final inverse estimates of all trials with this procedure. For each target ψ_k^*, the initial inverse estimate $q^* = g(\psi_k^*)$ was sent to the length controller and was active for 5 s before the feedback control was activated. Then, the feedback control on top of g was applied with 5 Hz and gain $\alpha = 0.02$ for 15 s, so that the overall evaluation time per target was 20 s, consistently with other evaluations in this paper. Results for the first trial are shown in Fig. 9: the heavy-tail in the error histogram has disappeared [compare Fig. 7(c)] and the maximum error is below 3 cm. The performance in 3-D shows an excellent match between the targets x_k^* (red) and actual positions x_k (blue).

Results for all three trials are shown in Table III (bottom). The mean Cartesian performance errors are reduced to 7–9 mm and the median errors to 6–8 mm, which is a substantial improvement and close to the accuracy baseline of 5 mm. While the amplitude of execution failures (not shown) is not reduced by the feedback control, the failure correlation has dropped to zero. No divergence of the feedback control was observed in the experiment. These results clearly show that the combination of a kinematically efficient inverse estimate that exploits all degrees of freedom, and a Cartesian feedback controller can cope with the problem of execution failures.

V. Nonstationary Behavior in Simulation

The experiments on the physical BHA have shown the success of our method for the robot's trunk morphology as well as physical problems like sensory noise or delayed execution of physical motions in time. Thereby, we have observed a significant change of the actuation ranges $\mathbf{Q}$. Other changes like drifting sensors or slight changes of the true forward function f due to viscoelasticity are known to occur but are hard to capture. This section complements the previous experiments with learning in a simulated environment in which such nonstationary behaviors can be controlled, and their effect on the learning method can be effectively investigated.

A. Kinematic Simulation of the BHA

To simulate the kinematics of the BHA we use an open source implementation [31] of a constant curvature continuum kinematics model. This model assumes that bending and stretching movements of each robot segment behave like a torus section, which allows to infer the coordinate transformations for the forward kinematics. The model allows to predict the end-effector position x of the BHA based on the actuator lengths q with an average accuracy of 1 cm [31]. Instead of applying a length on the robot, the end-effector position is simply computed with this library: $x = f^{\text{sim}}(q)$.

An important aspect of the BHAs kinematics are the actuation ranges $\mathbf{Q}$. Since this set is also not known analytically we used the minimum/maximum pressure results recorded on the real BHA (see Table I). The eight combinations of minimum/maximum pressure were recorded for each segment separately. The possible length combinations $\mathbf{Q}^{\text{sim}}_{(i)} \subset \mathbb{R}^3$ for a segment i are modeled by the convex hull of the resulting eight lengths. The possible lengths of different segments are modeled independently: $\mathbf{Q}^{\text{sim}} = \mathbf{Q}^{\text{sim}}_{(1)} \times \mathbf{Q}^{\text{sim}}_{(2)} \times \mathbf{Q}^{\text{sim}}_{(3)}$. When the exploration suggests a posture $q^* \notin \mathbf{Q}^{\text{sim}}$, it is projected onto the surface of $\mathbf{Q}^{\text{sim}}$

$$q = c(q^*) = \begin{cases} q^*, & \text{if } q^* \in \mathbf{Q}^{\text{sim}} \\ \underset{\hat{q} \in \mathbf{Q}^{\text{sim}}}{\text{argmin}} ||q^* - \hat{q}||, & \text{else.} \end{cases}$$

B. Nonstationary Actuation Ranges

We first investigate varying actuation ranges $\mathbf{Q}^{\text{sim}}$, which have been identified as important problem of the physical robot. The experiments in Section IV-C have shown that a feedback controller on top of learning can deal with a certain amount of residual errors caused by this problem. Yet, it is impossible to quantify on the actual robot to what extent the learning can actually deal with these changes because there is no accessible baseline: the robot cannot be made stationary to compare learning with and without changes of the system. Simulation experiments allow to manipulate nonstationary behavior and to quantify its impact.

For a direct comparison with the real BHA results, we performed three independent trials, each with $T = 90\,000$ examples and parameters identical to the previous experiments. We initially let the learning run on a stationary system for $T_{(s)} = 45\,000$ examples. Between $T_{(s)} = 45\,000$ and $T = 90\,000$, we reduced the ranges of two actuators continuously. Both the minimum and maximum values are narrowed by 30% for the first actuator of segment 2 and the second actuator of segment 3. The progress was linear in t. We investigated how the learning procedure can deal with this change, as well as how the performance develops if learning is stopped at the onset of nonstationary behavior. Results are shown in Fig. 10(a). Ongoing learning reduces the performance error even after the onset of change. When learning is turned off, the error increases slowly. While the increase of the average error is comparably mild, there are drastic differences in the maximum errors over the K targets in $\mathbf{X}^*$: the first simulated trial exposes a maximum error of 5.5 cm after $T = 90\,000$. When learning is turned off, the same trial results in 10% of the target positions with more than 10-cm error (maximum 20 cm). During the ongoing change of the ranges, learning is able to continuously find new solutions to reach for goals, once previously learned solutions become unreachable.

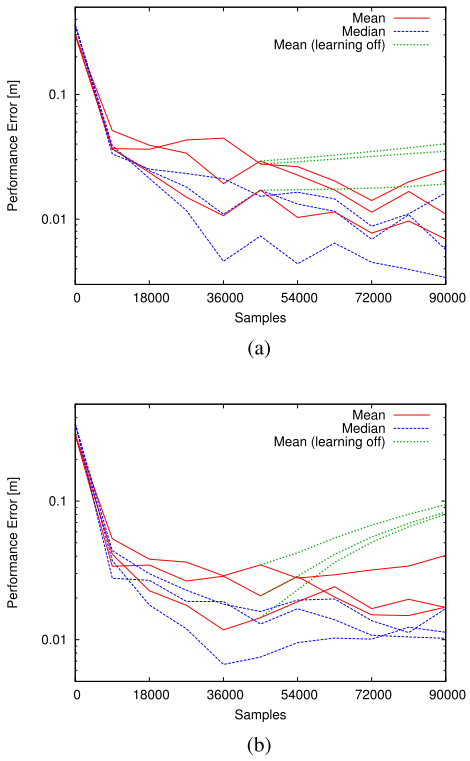


Fig. 10. Performance for (a) shrinking ranges and (b) for drifting sensors.

C. Sensory Drifts and Morphological Growth

We finally investigate different kinds of nonstationary behaviors for which there is no direct or quantitative evidence on the BHA, but which demonstrate the generality of our approach. A kind of nonstationary behavior that is typical for robotic systems is the drift of sensor values, when the physical sensors are not repeatedly calibrated. Such behavior is plausible, but hard to quantify, for the BHAs pressure and length sensors. We model such behavior in the BHA simulation by defining a drift function $d : \mathbb{R}^9 \rightarrow \mathbb{R}^9$ that

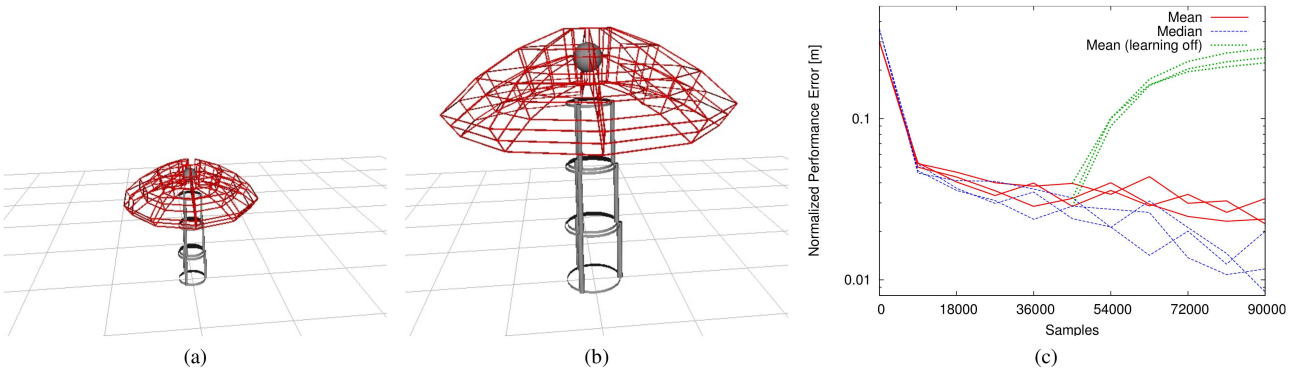


Fig. 11. We simulated morphological growth of the BHA (a) from half its size (b) to its full scale. (c) Without learning the error increases rapidly.

distorts the measurements of the actuator-lengths

$$d(q) = (\mathbb{1}_9 + \beta \cdot \mathtt{diag}(\vec{s})) \cdot q + \beta \cdot \vec{o}$$

where $\vec{s}$ and $\vec{o}$ are a linear distortion. β allows to scale its impact. When the learner operates with a length q, the true lengths with respect to effector position and ranges are $d(q)$

$$f'(q) = f^{\text{sim}}(d(q)), \quad c(q^*) = d^{-1}\left(c\left(d(q^*)\right)\right).$$

Again, we simulated three trials over $T = 90\,000$, with a sensory drift beginning at $T_{(s)} = 45\,000$. The entries of $\vec{s}$ and $\vec{o}$ where drawn from a normal distribution with deviation 0.05 independently for each trial. The drift amplitude β was linearly scaled from 0.0 to 1.0 between $T_{(s)}$ and T. Results are shown in Fig. 10(b). Without learning, the performance error increases significantly and reaches a average level of 8–10 cm. With enabled learning the performance error is approximately stabilized, although the amplitude and rate of the drift is too strong to further reduce the error as in the previous experiment.

The last experiment deals with a nonstationary behavior that is, in particular in its amplitude, clearly not a problem on the real BHA. It shows that our method can deal with even more drastic changes as they occur in infant development, which served as inspiration for our approach. We perform learning on a growing simulation of the BHA. The simulation starts with a BHA that is scaled to half of its original size and grows to full size between $T_{(s)} = 45\,000$ and $T = 90\,000$ [see Fig. 11(a) and (b)]. The change goes on linearly and concerns the radius of the simulated segments, the actuation ranges, as well as the reachable workspace. To assess the learning performance for a workspace with varying size the resulting errors are normalized to $(1/\gamma)E^X$, where $\gamma \in [0.5; 1.0]$ is the current relative size of the simulated BHA. The results [Fig. 11(c)] show that the performance without learning degenerates to almost initial error values. With enabled learning the median error is nevertheless decreasing, while the mean error is approximately constant. Since also the goals grow with the robot, the learning procedure has to continuously discover new goals on the top surface of the target volume. Based on the continuous sampling of the workspace (see Fig. 6) the learner has to initially make extrapolations how to reach them. These extrapolations are not necessarily perfect (as shown by the increased error in the learning off condition), but on a short distance (during slow change) good enough to trigger an efficient reexploration and therefore learning.

This experiment generates the largest gap between the learning and nonlearning during nonstationary behavior. Although the morphological change extinguishes the learned performance when learning is turned off, the change seems to be comparably easy to track during learning. This result clearly shows how goal-directed exploration also contributes to the successful mastery of infants' learning during growth.

VI. Conclusion

We have shown that online goal babbling allows to bootstrap the inverse kinematics of the pneumatically actuated BHA, which provides the first quantitative proof for the success of goal-directed bootstrapping schemes in real-world scenarios. The method is robust enough to cope with the inherent sensory noise, delays during the execution, and the varying actuator ranges. The successful learning of reaching skills is an important milestone for the applicability of such systems in practical real world scenarios. For the BHA, no control concept could so far be demonstrated that can deal with the platform's substantial challenges. Hence, our learning approach, together with a novel integration of feedback control, allows for the first time the practical operation of this system in terms of an accurate and reliable positioning of its end-effector. The method is, thereby fast enough to perform on the robot in reasonable time. We used 90 000 samples during our experiments, which corresponds to approximately 1000 crossings of the Cartesian workspace. Previous methods for the learning of inverse models in high dimensions required, even on much simpler and stationary 2-D problems, up to ten thousands or hundred thousands of such movements [12], [25], or several million samples for simple 3-D problems [21], which is not practical on a real robot.

The learned skill represents a direct, feedforward control from desired effector position to actuator lengths. Residual inaccuracies are unavoidable for feedforward control schemes. Yet, we have shown that such errors can be handled with an additional Cartesian feedback controller if necessary. The controller exploits the learner's efficient use of all actuators, which even allows to correct errors that are caused by the

narrow actuation ranges. The use of feedforward control is highly beneficial for a pneumatic robot: delays usually only allow to apply feedback-control with very low gains, which implies slow movements. A feedforward controller can quickly estimate the necessary motor commands, which can be applied immediately. This is particularly useful due to the narrow actuation ranges, for which the learned model has already stored valid solutions while a pure feedback controller needs to search for them newly during each movement.

Besides learning on the nonstationary robot, we have shown in simulation how the method copes with various changes such as changing ranges, drifting sensors, and even morphological growth. For each of these setups, we have shown that the performance degenerates significantly without learning, but is stable or improves for ongoing learning. The key idea to master such changes on a high-dimensional motor system is to structure exploration in a goal-directed manner. Goal babbling defines an incremental and ongoing process that is always based on currently observed data, and thus grounded on the current system behavior. Most importantly, it does not require an exhaustive exploration of the motor system. This could not even be done once on robots with many degrees of freedom like the BHA, so that a tracking of ongoing changes would even conceptually not be possible. Goal babbling discards redundant choices if multiple motor commands exist to solve the same target position, although additional mechanisms allow to exploit multiple solutions as well [32]. Hence, it only samples a low-dimensional submanifold in the space of motor commands, which can be quickly explored. Online learning then quickly reacts to a changing environment and allows to adapt to changes efficiently.

REFERENCES

[1] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: A survey," *Cognit. Process.*, vol. 12, no. 4, pp. 319–340, Nov. 2011.

[2] C. Laschi, B. Mazzolai, V. Mattoli, M. Cianchetti, and P. Dario, "Design of a biomimetic robotic octopus arm," *Bioinspiration Biomimetics*, vol. 4, no. 1, p. 015006, Mar. 2009.

[3] A. Grzesiak, R. Becker, and A. Verl, "The bionic handling assistant—A success story of additive manufacturing," *Assembly Autom.*, vol. 31, no. 4, pp. 329–333, 2011.

[4] K. Hosoda, S. Sekimoto, Y. Nishigori, S. Takamuku, and S. Ikemoto, "Anthropomorphic muscular-skeletal robotic upper limb for understanding embodied intelligence," *Adv. Robot.*, vol. 26, no. 7, pp. 729–744, 2012.

[5] D. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends Cognit. Sci.*, vol. 2, no. 9, pp. 338-347, Sep. 1998.

[6] P. Gaudiano and D. Bullock, "Vector associative maps unsupervised real-time error-based learning and control of movement trajectoires," *Neural Netw.*, vol. 4, no. 2, pp. 147–183, 1991.

[7] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proc. IEEE IROS*, Oct. 2001, pp. 298–303.

[8] Y. Demiris and A. Dearden, "From motor babbling to hierarchical learning by imitation: A robot developmental pathway," in *Proc. EpiRob*, 2005, pp. 31–37.

[9] R. Martinez-Cantin, M. Lopes, and L. Montesano, "Body schema acquisition through active learning," in *Proc. IEEE ICRA*, May 2010, pp. 1860–1866.

[10] A. Baranes and P.-Y. Oudeyer, "Robust intrinsically motivated exploration and active learning," in *Proc. IEEE 8th ICDL*, Jun. 2009, pp. 1–6.

[11] C. von Hofsten, "Eye–hand coordination in the newborn," *Develop. Psychol.*, vol. 18, no. 3, pp. 450–461, May 1982.

[12] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Trans. Auto. Mental Develop.*, vol. 2, no. 3, pp. 216–229, Sep. 2010.

[13] N. Bernstein, *The Co-ordination and Regulation of Movements*. New York, NY, USA: Pergamon, 1967.

[14] D. Bullock, S. Grossberg, and F. H. Guenther, "A self-organizing neural model of motor equivalent reaching and tool use by a multijoint arm," *J. Cognit. Neurosci.*, vol. 5, no. 4, pp. 408–435, 1993.

[15] M. Kawato, "Feedback-error-learning neural network for supervised motor learning," in *Advanced Neural Computers*. New York, NY, USA: Elsevier, 1990.

[16] M. Jordan and D. Rumelhart, "Forward models: Supervised learning with distal teacher," *Cognit. Sci.*, vol. 16, no. 3, pp. 307–354, Jul. 1992.

[17] J. Peters and S. Schaal, "Reinforcement learning by reward-weighted regression for operational space control," in *Proc. ICML*, 2007, pp. 745–750.

[18] A. Baranes and P.-Y. Oudeyer, "Maturationally-constrained competence-based intrinsically motivated learning," in *Proc. ICDL*, 2010, pp. 1–7.

[19] M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *Proc. IEEE ICDL*, Aug. 2011, pp. 1–8.

[20] U. Sailer, J. R. Flanagan, and R. S. Johansson, "Eye–hand coordination during learning of a novel visuomotor task," *J. Neurosci.*, vol. 25, no. 39, pp. 8833–8842, Sep. 2005.

[21] L. Jamone, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Learning task space control through goal directed exploration," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 702–708.

[22] P. O. Stalph and M. V. Butz, "Learning local linear Jacobians for flexible and adaptive robot arm control," *Genet. Program. Evolvable Mach.*, vol. 13, no. 2, pp. 137–157, Jun. 2012.

[23] C. Hartmann, J. Boedecker, O. Obst, S. Ikemoto, and M. Asada, "Real-time inverse dynamics learning for musculoskeletal robots based on echo state Gaussian process regression," in *Proc. RSS*, 2012, pp. 1–8.

[24] M. Rolf, J. J. Steil, and M. Gienger, "Mastering growth while bootstrapping sensorimotor coordination," in *Proc. ICER*, Sep. 2010, pp. 1–8.

[25] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robot. Autonom. Syst.*, vol. 61, no. 1, pp. 49–73, 2013.

[26] (2010). *Deutscher Zukunftspreis (German Future-Award)* [Online]. Available: http://www.deutscher-zukunftspreis.de/en/content/2010

[27] K. Neumann, M. Rolf, and J. J. Steil, "Reliable integration of continuous constraints into extreme learning machines," in *Proc. Int. Symp. Extreme Learn. Mach.*, 2013, pp. 1–6.

[28] (2013). *VICON Motion Tracking Systems* [Online]. Available: http://www.vicon.com

[29] T. D. Sanger, "Failure of motor learning for large initial errors," *Neural Comput.*, vol. 16, no. 9, pp. 1873–1886, Sep. 2004.

[30] H. Ritter, "Learning with the self-organizing map," in *Artificial Neural Networks*, T. Kohonen, Ed. New York, NY, USA: Elsevier, 1991.

[31] M. Rolf and J. J. Steil, "Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant," in *Proc. IEEE/RJS IROS*, Oct. 2012, pp. 3440–3446.

[32] F. R. Reinhart and M. Rolf, "Learning versatile sensorimotor coordination with goal babbling and neural associative dynamics," in *Proc. IEEE ICDL*, Jun. 2013, pp. 327–332.

Matthias Rolf received the master's (Hons.) degree in computer science and the Ph.D. degree (*summa cum laude*) in engineering from Bielefeld University, Bielefeld, Germany, in 2008 and 2012, respectively. His dissertation was titled "Goal Babbling for an Efficient Bootstrapping of Inverse Models in High Dimensions."

He was a Research Fellow with the Research Institute for Cognition and Robotics, Bielefeld University, from 2008 to March 2013, receiving a scholarship from Bielefeld University and Honda Research Institute Europe from 2008 to 2011. Since 2013, he has been a specially appointed Researcher with the Emergent Robotics Laboratory, Osaka University, Osaka, Japan. He investigated goal-directed exploration schemes mimicking human neonates' exploratory behavior for an efficient learning of sensorimotor skills in robotics systems. His current research interests include developmental robotics, machine learning, software engineering, and attention systems.

Jochen J. Steil received the Diploma degree in mathematics and the Ph.D. degree in computer science from the University of Bielefeld, Bielefeld, Germany, in 1993 and 1999, respectively. His dissertation was titled "Input–Output Stability of Recurrent Neural Networks." He received the venia legendi in neuroinformatics in 2006.

He was with St. Petersburg Electrotechnical University, St. Petersburg, Russia, from 1995 to 1996, supported by a German Academic Exchange Foundation Grant. He was with the Honda Research Institute Europe, Offenbach, Germany, as a Principal Scientist. Since 2007, he has been a Managing Director of the Institute for Cognition and Robotics, Bielefeld, as a Scientific Board Member of the cognitive interaction technology excellence cluster and has been an Außerplanmäßiger Professor of neuroinformatics with the Faculty of Technology since 2008. From 2010 to 2014, he coordinated the FP7-IP AMARSI-Adaptive Modular Architectures for Rich Motor Skills. He heads several industry-transfer projects within the German leading edge cluster intelligent technical systems in Ost-Westphalia. His current research interests include dynamical systems and recurrent networks, learning architectures, neurorobotics, motion representation, kinesthetic teaching, and modeling of attention in robotics.