Multi-Stage Goal Babbling for Learning Inverse Models Simultaneously

Rania Rayyes, Daniel Kubus and Jochen Steil

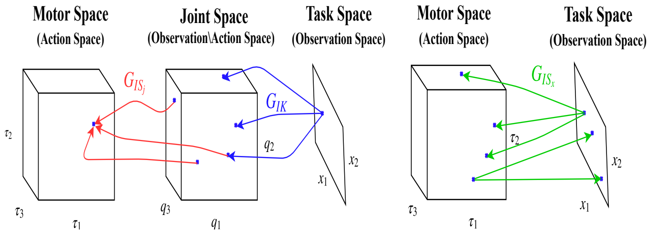
***Abstract*— Online Goal Babbling has been proposed for directed learning of inverse kinematics from scratch following the paradigm of ”learning while behaving”. In this paper, we show that Goal Babbling can be modified to learn inverse statics – primarily for gravity compensation – online, from scratch and in a plain exploratory fashion. We primarily focus on devising a Multi-Stage Goal Babbling for learning inverse kinematics and inverse statics simultaneously. The results are quite promising and clearly demonstrate that the achieved accuracy is more than sufficient for most handling tasks.**

I. INTRODUCTION

Direct learning of inverse models for advanced robots has been proposed to deal with lack of prior knowledge and inaccurate models, e.g. [1]. Learning from scratch without requiring prior knowledge is at the core of Goal Babbling (GB) [2] which has been proposed to mimic infants learning of motor skills, i.e. reaching (learning how to reach by trying to reach and updating the motion by iterating the trials). GB has been first implemented for direct learning of inverse kinematics (IK) [2]. It has been also adopted in other domains owing to its high flexibility, e.g. generating speech [3], [4], avoiding obstacles [5] and using tool [6]. We propose in this paper a constrained GB to learn inverse statics (IS) for gravity compensation online, from scratch, and without using a feedback controller in a plain exploratory fashion, whereas the previous research on learning IS has been done offline only using a closed-loop controller to collect training data and often enhanced already existing (parametric) models (e.g., [7]–[9]). We also propose multi-stage GB to learn IK and IS simultaneously. We demonstrate the results for 2R and 3R arms only because GB has already demonstrated high scalability up to 9-DoF floating-base compliant humanoid (COMAN) [10] and up to 50-DoF planar arm [2]. Therefore, learning IK and IS which map from the lower dimensional task space to the higher dimensional motor space can both be scaled very well. However, scaling GB for learning IS which maps from the joint space to the motor space might be difficult for higher DoFs as both spaces scale with the number of DoFs.

II. MULTI-STAGE GOAL BABBLING

We will first introduce the original GB approach and then explain our proposed constrained GB and multi-stage GB.



(a) G_{IK} and G_{IS_j} maps (b) G_{IS_x} map

Fig. 1. Multi-stage goal babbling maps

A. Online Goal Babbling

GB has been proposed to learn the inverse kinematics map $\boldsymbol{G}_{IK}$ (cf. Fig. 1(a)) that assigns to each end-effector position $\boldsymbol{p} \in \mathcal{P}_a \subset \mathbb{R}^n$ the corresponding configuration $\boldsymbol{q} \in \mathcal{Q}_p \subset \mathbb{R}^m$ that is required to attain it, where m is the number of DoFs and n is the dimension of the target variable (e.g. $n \in \{2, 3\}$ for the spatial position of the end-effector):

$$\boldsymbol{G}_{IK} : \mathcal{P}_a \rightarrow \mathcal{Q}_p \tag{1}$$

$\mathcal{P}_a$ is the set of attainable positions in task space and $\mathcal{Q}_p$ is the set of permissible configurations in joint space. $\boldsymbol{G}_{IK}$ is learned by exploring actions (configurations) and observing their outcomes (end-effector positions). The observation space represents the task space and the action space represents the joint space. Correlated exploratory noise is added to the motor commands to discover and learn novel outcomes. These samples are generated by means of goal-directed movement attempts [2]. Local Linear Map (LLM) [11] is employed because an incremental regression mechanism is required to update the inverse estimates online.

B. Constrained Online Goal Babbling for Learning IS

We define the set of statically admissible torques (SST):

$$\mathcal{T}_s = \{\boldsymbol{\tau} | \exists \boldsymbol{q} \in \mathcal{Q}_p : \boldsymbol{\tau} - \boldsymbol{G}(\boldsymbol{q}) = 0\} \tag{2}$$

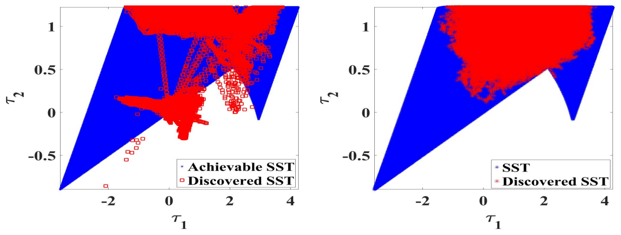
as the set $\mathcal{T}_s$ of all torque vectors required to maintain the configurations in the set $\mathcal{Q}_p$ in the static equilibrium case, i.e. $\dot{\boldsymbol{q}} = \boldsymbol{0}$ and $\ddot{\boldsymbol{q}} = \boldsymbol{0}$. We aim to learn the inverse statics maps $\boldsymbol{G}_{IS_x}$ and $\boldsymbol{G}_{IS_j}$:

$$\boldsymbol{G}_{IS_x} : \mathcal{P}_a \rightarrow \mathcal{T}_s, \qquad \boldsymbol{G}_{IS_j} : \mathcal{Q}_p \rightarrow \mathcal{T}_s \tag{3}$$

As shown in Fig. 1, $\boldsymbol{G}_{IS_x}$ maps from the lower dimensional task space to the higher dimensional motor space and assigns to each position $\boldsymbol{p} \in \mathcal{P}_a \subset \mathbb{R}^n$ the required static torque $\boldsymbol{\tau} \in \mathcal{T}_s \subset \mathbb{R}^m$ to maintain it. $\boldsymbol{G}_{IS_j}$ maps from the joint space to the motor space and assigns to each valid configuration

The authors are with Technische Universität Braunschweig, Institut für Robotik und Prozessinformatik, 38106 Braunschweig, Germany {rra,dku,jsteil}@rob.cs.tu-bs.de

Workshop on BODIS: The utility of body, interaction and self learning in robotics
International Conference on Intelligent Robots IROS 2018, Madrid, Spain



(a) with original GB

(b) with constrained GB

Fig. 2. Discovered SST for the 2R planar arm

$\boldsymbol{q} \in \mathcal{Q}_p \subset \mathbb{R}^m$ the required static torque $\boldsymbol{\tau} \in \mathcal{T}_s \in \mathbb{R}^m$ to maintain it.

In order to learn IS with GB, some constraints should be considered. Learning IS in a plain exploratory fashion is challenging since not all combination of joint torques result in an admissible torque. Adding exploratory noise to motor commands (torques) in GB may results in inadmissible torque, i.e. invalid torque which inevitably accelerates the robot and make it hit its joint limits[1]. Applying such a torque bears the risk to damage the robot if no further safety precautions are taken. Moreover, the learner will be disturbed because of the resulting invalid training sample (i.e. inadmissible torque which does not correspond to the joint limit configuration where the robot settles in). This imposes limits on torque combinations in addition to joint-wise torque limits. Hence, SST should be estimated beforehand or learned, when the robot hits its joint limits during exploration, the corresponding torque is considered inadmissible and the SST estimate is updated accordingly. Delaunay triangulation is used to estimate SST boundary and the nearest neighbor algorithm is employed to assign each invalid torque to a valid one before execution. Fig. 2 illustrates the SST for a 2R planar arm with specific joint limits. Fig. 2(a) shows that applied torques may not be contained in the SST (red dots outside the SST) due to adding exploratory noise in GB. The SST could be correctly explored as shown in Fig. 2(b) by applying constrained GB.

C. Multi-Stage Goal Babbling

We aim in this paper to learn $\boldsymbol{G}_{IK}$ (cf. (1)), $\boldsymbol{G}_{IS_j}$ and $\boldsymbol{G}_{IS_x}$ (cf. (3)) simultaneously. In other words, our goal is to learn the required configurations and the corresponding static torques to attain and maintain some predefined position targets $\mathcal{X}^* \subset \mathcal{P}_a$. In multi-stage GB as illustrated in Fig. 1, there are two observation spaces: the task space which represents end-effector positions and the joint space which serves as an observation space for G_{IS_j} and as an action space for G_{IK}. The motor space (action space) represents motor commands (torques). G_{IK} is a one-to-many mapping i.e. each position can be reached by different configurations, whereas G_{IS_j} is a many-to-one mapping, i.e. multiple configurations require the same static torque to be maintained. Clearly, G_{IS_x} is a many-to-many mapping i.e. each position can be maintained by multiple torques, and multiple positions can be maintained by the same torque as illustrated in Fig. 1(b).

Algorithm 1 Multi-Stage Goal Babbling

```
procedure MSGB($\boldsymbol{x}^{home}$,$\boldsymbol{q}^{home}$, $\boldsymbol{\tau}^{home}$, $\mathcal{X}^*$, $SST$)
    INITLLM( $\boldsymbol{x}^{home}$, $\boldsymbol{q}^{home}$, $\boldsymbol{\tau}^{home}$)
    for $N$ number of iterations do
        for each randomly chosen target $\boldsymbol{x}^* \in \mathcal{X}^*$ do
            for $l = 1 : L$ do
                generate an intermediate target $\boldsymbol{x}_t^*$
                estimate configuration value $\hat{\boldsymbol{q}}_t^*$ for $\boldsymbol{x}_t^*$
                    $\hat{\boldsymbol{G}}_{IK}(\boldsymbol{x}_t^*) = \hat{\boldsymbol{q}}_t^*$
                estimate torque value $\hat{\boldsymbol{\tau}}_t^*$ for $\hat{\boldsymbol{q}}_t^*$
                    $\hat{\boldsymbol{G}}_{IS_j}(\hat{\boldsymbol{q}}_t^*) = \hat{\boldsymbol{\tau}}_t^*$
                add exploratory noise $\boldsymbol{\sigma}$:
                    $\boldsymbol{\tau}_t^+ = \hat{\boldsymbol{\tau}}_t^* + \boldsymbol{\sigma}(\boldsymbol{q}_t^*, t)$
                if $\boldsymbol{\tau}_t^+ \notin \boldsymbol{SST}$ then
                    $\boldsymbol{\tau}_t^+ = \boldsymbol{\tau}_m$ where
                        $\{\boldsymbol{\tau}_m \in \mathcal{T}_s : \forall \boldsymbol{\tau}_n \in \mathcal{T}_s$
                        $dist(\boldsymbol{\tau}_t^+, \boldsymbol{\tau}_m) \leqslant dist(\boldsymbol{\tau}_t^+, \boldsymbol{\tau}_n)\}$
                end if
                execute $\boldsymbol{\tau}_t^+$ and observe $(\boldsymbol{q}_t^+, \boldsymbol{x}_t^+)$
                compute weight $w_t$
                TRAINLLM($\boldsymbol{\tau}_t^+, \boldsymbol{q}_t^+, \boldsymbol{x}_t^+, w_t$)
            end for
        end for
    end for
end procedure
```

Algo.1 illustrates the multi-stage GB steps. At the beginning, each initial inverse estimate suggests some default value at time instant $t = 0$, i.e. some comfortable posture $\boldsymbol{q}^{home}$ or default torque $\boldsymbol{\tau}^{home}$ corresponding to the home position x^{home} (i.e. the initial position):

$$\left.\begin{aligned} \hat{\boldsymbol{G}}_{IK}(\boldsymbol{x}) &= \boldsymbol{q}^{home} \\ \hat{\boldsymbol{G}}_{IS_x}(\boldsymbol{x}) &= \boldsymbol{\tau}^{home} \\ \hat{\boldsymbol{G}}_{IS_j}(\boldsymbol{q}) &= \boldsymbol{\tau}^{home} \end{aligned}\right\} \tag{4}$$

For a selected number of iterations and a set of predefined position targets $\mathcal{X}^*$, the targets in $\mathcal{X}^*$ are chosen randomly and iteratively. The starting position $\boldsymbol{x}^{home}$ will be selected as a target with a probability of p^{home} to avoid drifting. Continuous linear paths of L intermediate targets $\boldsymbol{x}_t^*$ are generated iteratively by interpolating between the current predefined target and the next chosen one. The robot tries to reach each target $\boldsymbol{x}_t^*$ as follows:
The current inverse kinematics estimate $\hat{\boldsymbol{G}}_{IK}(\boldsymbol{x}) = \hat{\boldsymbol{q}}_t^*$ is used as a joint target for the inverse static estimate $\hat{\boldsymbol{G}}_{IS_j}(\hat{\boldsymbol{q}}_t^*) = \hat{\boldsymbol{\tau}}_t^*$ which is used as a motor torque, and correlated exploratory noise $\boldsymbol{\sigma}$ [2] is added to the estimated torque $\hat{\boldsymbol{\tau}}_t^*$ (5) in order to discover and learn novel outcomes:

$$\boldsymbol{\tau}_t^+ = \hat{\boldsymbol{\tau}}_t^* + \boldsymbol{\sigma}(\boldsymbol{x}_t^*, t) \tag{5}$$

where t is the time step, and $\boldsymbol{\tau}_t^+$ is the torque which is applied to the robot if statically admissible. Otherwise, it

[1] Obviously, in practical applications, a software joint limit is employed to avoid reaching the hardware joint limit.

will be assigned to the nearest neighbor in the SST and applied. The explored motor command $\boldsymbol{\tau}_t^+$ and the resulting outcomes $(\boldsymbol{q}_t^+, \boldsymbol{x}_t^+)$ are observed and used as supervised learning example to update the inverse estimates immediately before the next intermediate target is generated. Executing $\boldsymbol{\tau}_t^+$ will result in $\boldsymbol{q}_t^+$ and the corresponding torque estimated by the learner for $\boldsymbol{q}_t^+$ is denoted by $\hat{\boldsymbol{\tau}}_t^+$. Similarly, $\boldsymbol{q}_t^+$ will result in $\boldsymbol{x}_t^+$ and the corresponding estimated configuration for $\boldsymbol{x}_t^+$ is denoted by $\hat{\boldsymbol{q}}_t^+$. The goal is thus to minimize the error E_t^q between the real and the estimated configurations and the error E_t^τ between the applied and the estimated torques to improve the estimation accuracy:

$$E_t^q = w_t \|\boldsymbol{q}_t^+ - \hat{\boldsymbol{q}}_t^+\|^2, \qquad E_t^\tau = w_t \|\boldsymbol{\tau}_t^+ - \hat{\boldsymbol{\tau}}_t^+\|^2 \tag{6}$$

Output torque sensors are desirable but not crucial for our methods. Either the commanded torques or the measured output torques can be used in our proposed scheme.

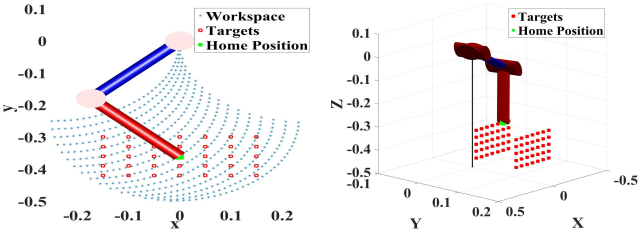
1) Weighting Sample Scheme: Multi-stage GB tries to select the most efficient solution to handle redundancy and avoid inconsistent samples (e.g. the same end-effector position but different joint angles and torques) by using the following weighting scheme:

$$\left.\begin{aligned} w_t^{dir} &= \frac{1}{2}(1 + cos\sphericalangle(\boldsymbol{q}_t^* - \boldsymbol{q}_{t-1}^*, \boldsymbol{q}_t^+ - \boldsymbol{q}_{t-1}^+) \\ w_t^{eff} &= \| \boldsymbol{x}_t^+ - \boldsymbol{x}_{t-1}^+ \| \cdot \| \boldsymbol{q}_t^+ - \boldsymbol{q}_{t-1}^+ \|^{-1} \\ w_t &= w_t^{dir} \cdot w_t^{eff} \end{aligned}\right\} \tag{7}$$

The direction criterion w_t^{dir} assesses whether the observed direction and the planned one align well. The efficiency criterion w_t^{eff} assesses the movement efficiency. The ambiguity of the mappings i.e. the one-to-many mapping $\boldsymbol{G}_{IK}$ (redundancy resolution) and the many-to-one mapping $\boldsymbol{G}_{IS_j}$ (torque ambiguity) as well as the many-to-many mapping $\boldsymbol{G}_{IS_x}$ are controlled by the home posture $\boldsymbol{q}^{home}$. $\boldsymbol{q}^{home}$ is used not only as initial configuration but also as returning point in order to avoid drifting and determine outcomes. By using the weighting scheme, only one solution for the joint configuration and the corresponding torque and end effector position will be learned which is controlled by the home posture as well [12]. Although due to torque ambiguity the preferable configuration might not be guaranteed and the robot might settles in a different one, still the algorithm demonstrates robust performance without inconsistencies.

III. Experimental Results

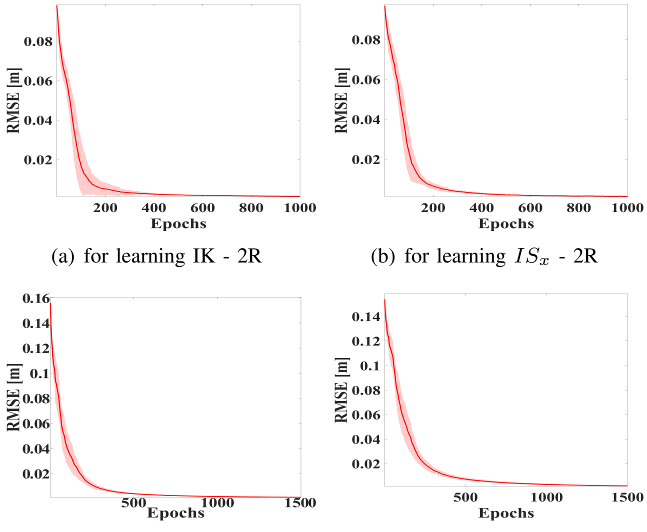
We present results for learning IK and IS simultaneously with multi-stage GB for a 2R planar arm and a 3R simplified human arm [13] shown in Fig. 3(a) and Fig. 3(b) respectively. The models have been set up in MATLAB using Robotics Toolbox [14]. Parameter optimization has been done using pattern search approach [15] to obtain the set of GB parameters shown in Table. I in order to minimize the training errors E_t^q and E_t^τ and speed up exploration.



(a) 2R planar arm

(b) 3R spatial arm

Fig. 3. Training Targets and Robot models with 0.25 m link length



(a) for learning IK - 2R

(b) for learning IS_x - 2R

(c) for learning IK - 3R

(d) for learning IS_x - 3R

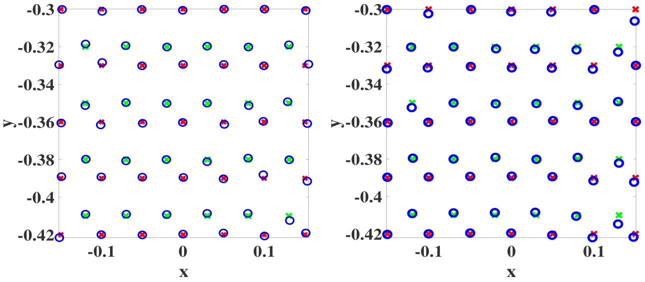
Fig. 4. Standard deviation for the robot performance error

A. Learning IK and IS for the 2R planar arm

The robot tries to reach and maintain 35 position targets regularly distributed on a grid (30×12 $[cm]$) as shown in Fig. 3(a). These targets are randomly and iteratively chosen during training with 7 intermediate targets generated between each two chosen ones (cf. Sec. II-C). The robot performance on all training targets is tested in each epoch, i.e. each 100 samples. The training error converges very fast as shown in Fig. 4. The robot reaches and maintains all the predefined targets as well as the intermediate targets very well with average root mean square error (RMSE) of 1.15 mm. The robot then tries to achieve 24 new position targets scattered on a regular grid, the testing performance was also very good illustrated with the blue circles in Fig. 5 with average RMSE of 1.1 mm. The robot then tries to achieve 24 new targets scattered in joint space as illustrated in Fig .7(a) in order to test the learned G_{IS_j}. Compared to the static torque limits $(-18.4, 24.5)$ Nm and $(-6, 6.2)$ Nm for the $1st$ and $2nd$ joints, the observed RMSE is negligibly small. Training and testing results are shown in Table. II.

B. Learning IK and IS for the 3R simplified human arm

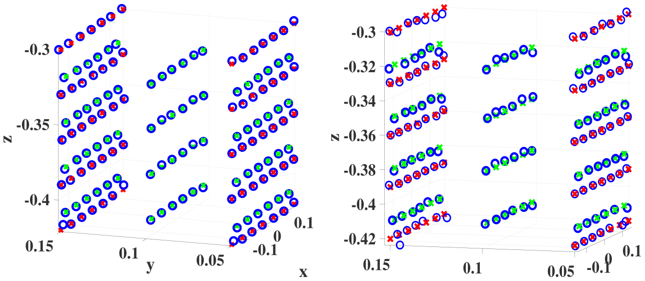
We did the same experiment with 70 position targets regularly distributed on two grids ($30 \times 12 \times 10$ $[cm]$) for the 3R arm (cf. Fig. 3(b)). Testing the robot performance



(a) G_{IK} 2R arm

(b) IS_x 2R arm

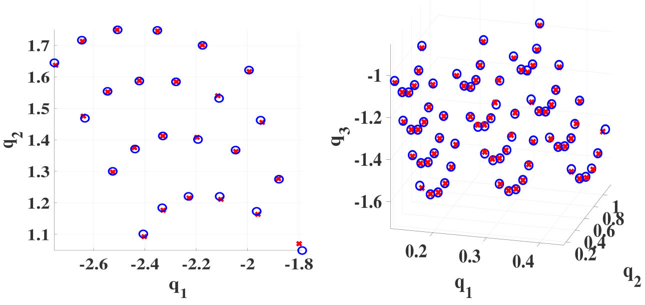
Fig. 5. Training and test performance for the 2R planar arm. The green dots represent the testing targets and the red ones represent the training targets. The blue circles represent the real end-effector positions.



(a) G_{IK} 3R arm

(b) IS_x 3R arm

Fig. 6. Training and test performance for the 3R arm. The green dots represent the testing targets and the red ones represent the training targets. The blue circles represent the real end-effector positions.



(a) for the 2R arm

(b) for the 3R arm

Fig. 7. Test performance of the learned G_{IS_j}. The red crosses represent the testing joint targets and the blue circles represent the real configurations.

has been done on 72 new targets scattered on three grids in Cartesian space and joint space as illustrated in Fig. 6 and Fig. 7(b) respectively. The training error converges very fast as shown in Fig. 4. The torque limits for the $1st$, $2nd$, and $3rd$ joints are $(-24.4, 24)$ Nm, $(-24.2, 24.2)$ Nm and $(-12.4, 12.2)$ Nm respectively. Again, the observed torque RMSE is negligible. The results are illustrated in Table. II.

IV. Outlook

If robot dynamics change e.g. adding a tool, the static model should be learned again from scratch. We therefore investigate scaling techniques for adapting the learned model online. Moreover, our recent work shows how to increase the learning efficiency by exploiting "Symmetries" i.e., the many-to-one characteristics of IS [16].

TABLE I

Multi-Stage GB Parameters

Robot Model	sigma σ	Intermediate Steps Nr.	Home Probability	Learning Rate
2R	0.75	7	0.03	0.07
3R	0.5	7	0.015	0.07

TABLE II

Multi-Stage GB Experimental Results for 2R and 3R arms

learned model 2R	Training RMSE			Testing RMSE		
	m	rad	Nm	m	rad	Nm
G_{IK}	10^{-3}	$57\cdot10^{-4}$		$8\cdot10^{-4}$	$48\cdot10^{-4}$	
G_{IS_j}		$6\cdot10^{-3}$	$44\cdot10^{-9}$		$65\cdot10^{-4}$	$27\cdot10^{-9}$
G_{IS_x}	$13\cdot10^{-4}$		$44\cdot10^{-9}$	$14\cdot10^{-4}$		$27\cdot10^{-9}$
learned model 3R	Training RMSE			Testing RMSE		
	m	rad	Nm	m	rad	Nm
G_{IK}	11.10^{-4}	$62\cdot10^{-4}$		$9\cdot10^{-4}$	$57\cdot10^{-4}$	
G_{IS_j}		$53\cdot10^{-4}$	$48\cdot10^{-9}$		$78\cdot10^{-4}$	$45\cdot10^{-9}$
G_{IS_x}	$15\cdot10^{-4}$		$48\cdot10^{-9}$	$15\cdot10^{-4}$		$45\cdot10^{-9}$

Acknowledgment

R. Rayyes receives funding from DAAD -"Research Grants-Doctoral Programme in Germany" scholarship.

References

[1] 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, pp. 1147–1160, 2014.

[2] M. Rolf *et al.*, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *IEEE ICDL*, 2011.

[3] A. K. Philippsen *et al.*, "Goal babbling of acoustic-articulatory models with adaptive exploration noise," in *ICDL*. IEEE, 2016, pp. 72–78.

[4] Moulin-Frier *et al.*, "Self-organization of early vocal development in infants and machines: The role of intrinsic motivation," *Frontiers in Psychology (Cognitive Science)*, vol. 4, no. 1006, 2013.

[5] P. Loviken and N. Hemion, "Online-learning and planning in high dimensions with finite element goal babbling," *IEEE ICDL*, 2017.

[6] S. Forestier and P. Oudeyer, "Modular active curiosity-driven discovery of tool use," in *IEEE/RSJ, IROS2016*, pp. 3965–3972.

[7] A. D. Luca and S. Panzieri, "Learning gravity compensation in robots: Rigid arms, elastic joints, flexible links," *Int. Journal of Adaptive Control and Signal Processing*, vol. 7, no. 5, pp. 417–433, 1993.

[8] M. Xie *et al.*, "Self learning of gravity compensation by loch humanoid robot," *Int. Conf. on Humanoid Robots*, 2008.

[9] M. Giorelli *et al.*, "Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 823–834, 2015.

[10] R. Rayyes and J. J. Steil, "Goal babbling with direction sampling for simultaneous exploration and learning of inverse kinematics of a humanoid robot," in *Proc. of the WS on NC2*, vol. 4, 2016, pp. 56–63.

[11] H. Ritter, "Learning with the Self-Organizing Map," in *ICANN-91*, T. Kohonen, Ed., vol. 1. North Holland, 1991, pp. 379–384.

[12] R. F. Reinhart and M. Rolf, "Learning versatile sensorimotor coordination with goal babbling and neural associative dynamics," in *IEEE ICDL*, Aug 2013, pp. 1–7.

[13] A. Babiarz *et al.*, "Dynamics modeling of 3d human arm using switched linear systems," *Asian Conf. on Intelligent Information and Database Systems*, vol. 9012, pp. 258–267, 2015.

[14] P. Corke, "A robotics toolbox for matlab," *IEEE RAM*, vol. 3, no. 1, pp. 24–32, March 1996.

[15] R. Lewis and V. Torczon, "Pattern search algorithms for bound constrained minimization," *SIAM Journal on Optimization*, vol. 9, no. 4, pp. 1082–1099, 1999.

[16] R. Rayyes *et al.*, "Learning inverse statics models efficiently with symmetry-based exploration," *To be published in Frontiers in Neurorobotics 2018.*