A Scalable Method for Multi-Stage Developmental Learning for Reaching

Wolfgang Burger*, Erhard Wieser*, Emmanuel Dean-Leon, and Gordon Cheng
Institute for Cognitive Systems
Technical University of Munich
Munich, Germany
Email: wolfgang.burger@tum.de, erhard.wieser@tum.de, dean@tum.de, gordon@tum.de

***Abstract*—In this paper, we introduce a technique to learn sensory-motor sequences in multiple consecutive stages, where one stage bootstraps sequences serving as training data for the subsequent stage. By introducing multiple interaction stages and recording the generated sensory-motor sequences of a preceding interaction stage, we obtain a system capable of self-generation of training data to increase the skill performance over time. At the beginning, our system uses a constrained degrees of freedom (DOF) exploration to gather a simple and short set of training data for a meaningful first-stage behavior. This minimum amount of samples already enables the robot to generate a reaching behavior for goals in the visual field. The generated observed sensory-motor sequences are then used as training data for a subsequent reaching phase. We are using the Predictive Action Selector (PAS) as a system building block, which provides bootstrapping of visual-proprioceptive predictions. Since our system was already presented on a robot with 2 DOF and 5 DOF, we proceed with the evaluation on a different robot with 6 DOF. Thus, we demonstrate the generality of the approach on various robotic platforms with different morphologies. By increasing the number of DOF, we continue showing the scalability of the presented system. Without any prior knowledge of neither the forward nor the inverse kinematics, the experiments show promising results with a reaching success rate of 66% during the first-stage reaching. This result is obtained by using only 13 training sequences (349 samples) which have been obtained during the constrained DOF exploration in only a few minutes. The developmental process is then shown by taking the generated sequences obtained during the first-stage reaching and using them as training data for the second-stage reaching. With second-stage reaching, the goal reaching times were reduced by up to 59% and, in contrast to first-stage reaching, it allows continuous retraining with increasing training data in subsequent stages.**

I. Introduction

Learning processes start early in human's lifetime and are facilitated by exploring the own body and the environment [1]. Focusing on the ability to reach objects in the visual field, we propose a system using the Predictive Action Selector (PAS) [2], [3] as main building block. Our system enables a robot to develop the skill of approaching selected goals in the visual field with the end-effector without any prior knowledge of the robot kinematics. The system is implemented and validated on the Tactile Omnidirectional Mobile Manipulator (TOMM) [4] (see fig. 1), while the PAS was originally presented on the Aldebaran NAO robot [2], [3].

*W. Burger and E. Wieser had an equal contribution to this paper. Video to the paper: https://youtu.be/okZz9HPRrZI

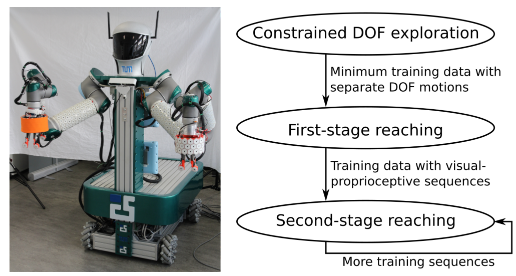

Fig. 1. Robot TOMM and a structural overview of the proposed developmental process. The system is self-generating training data for subsequent execution stages to improve reaching performance.

With the integration of the PAS onto the new platform, we show the general applicability of the method on different robotic platforms with altered robot kinematics. On TOMM, the number of degrees of freedom (DOF) increases from 5 DOF to 6 DOF, the camera setup allows stereo tracking, the visual field is larger and the reachable workspace increases with the size of the robot.

The proposed system uses multiple stages of development to increase the reaching performance over time. At the beginning, a constrained DOF exploration is used to gather a minimum amount of training data. Then, the system reaches for different static goals in the visual field by bootstrapping sequences from the obtained knowledge. For each goal, the system uses mental simulation to select and predict a sequence which is then executed and validated on the robot. During the first-stage reaching, new visual-proprioceptive sequences are recorded for the consecutive stages. After the first stage, the second stage uses the recorded observed sequences to compose new motion sequences for further goal reaching attempts. In sum, we present the different developmental stages available with the first-stage and second-stage mode of the PAS, and introduce its capability for mental simulation.

A. Related Work

In this section, we will present and distinguish some different approaches of autonomous development of reaching skills by two characteristics.

The first characteristic is the technique for the autonomous initial acquisition of training data which can be divided into three different approaches: motor babbling [5], [6], [7], [8],

[9], [10], goal babbling [11], [12], and constrained DOF exploration [2], [3]. While motor babbling (also known as body babbling) uses random DOF motions to record sequences of joint states and the corresponding visual features, goal babbling selects target goals in the task space and the system tries to reach them by using goal directed motions in the task space. Baranes and Oudeyer [13] show that for the learning of inverse models, goal babbling with well-chosen goals is able to generate sufficient data faster than motor babbling, which needs to explore the whole joint space. The third method also used in our approach is to execute a constrained DOF exploration which moves each DOF one after the other in the available directions with one DOF moving at a time, as presented by Wieser and Cheng in [2], [3]. Emulating the pre-structuring of the human biological motor system during the early developmental stage, the constrained DOF exploration is generating sufficient training data for an early sensory-motor mapping within a distinctively short time [3].

The second characteristic is the method of the knowledge acquisition and representation which is then used as forward or inverse model of the robot. Most of the approaches are different kinds of neural networks or purely mathematical models. Saegusa *et al.* [5] use a Multi Layer Perceptron (MLP) to approximate the kinematics of a robot by using the recorded visual-proprioceptive sequences obtained during active motor babbling. An early study by von Hofsten [14] analyzed the early development of pre-reaching and reaching in infants between the ages of 1 to 19 weeks. This behavior was reproduced on a humanoid iCub robot by Shaw *et al.* [8] using direct field mappings between joint states and spatial positions. Furthermore, the same study motivated Narioka and Steil [11] to model a similar U-shaped learning on a simulated robot using goal babbling with intrinsic motor noise to train Local Linear Maps (LLMs). Another approach using goal babbling is shown by Rayyes and Steil [12] to learn the robots inverse kinematic also using LLMs. Mahoor *et al.* [6] utilize autoencoder to create a map between joint space and visual space, and learn accurate and smooth goal reaching on a Meka Robotics M3 humanoid robot. Another research using autoencoders is presented by Luo *et al.* [9] where a PKU-HR6.0 humanoid robot develops reaching and grasping abilities. Srinivasa and Grossberg [7] use a self-organizing neural model consisting of multiple layers encoding different types of changes in the environment to generate saccades to goal positions in two different redundant simulated robotic systems. Dearden *et al.* [10] represent forward models with Bayesian networks to imitate presented actions.

Compared to the presented related work, the main contributions of our method are the following:

- Multi-stage developmental process with self-generation of training data for subsequent stages.
- Mental simulation of planned trajectories using a closed loop action selection system.
- General applicability on various robotic platforms with different morphology and different number of DOF.

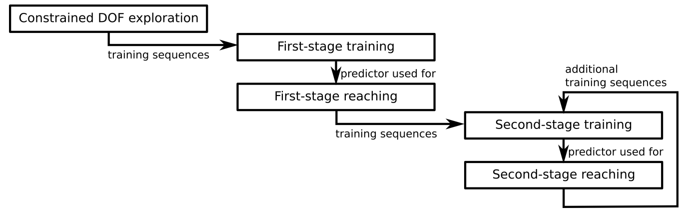

Fig. 2. Main phases during the developmental process. The arrows indicate training data or a trained predictor which is used for the consecutive phase.

B. Robot Description

The Tactile Omnidirectional Mobile Manipulator (TOMM) [4] is a human sized robot developed at the Institute for Cognitive Systems at the Technical University of Munich. The ten RGB cameras in the robot head are arranged in five stereo camera groups where each of them is facing in a different direction. For our experiments, we used one of the stereo camera pairs facing forward in a 45° angle towards the floor. We use stereo vision to obtain the three-dimensional visual feature vector representing the position of the end-effector in the robot's field of view using the coordinates in the left camera as v_1 and v_2, and the disparity as v_3. Finally, two UR5 industrial robots with 6 DOF are mounted on the left and right side of the torso. Since we are controlling 6 DOF for 3D visual feature goals, the robot can be considered redundant for the given task. There are several differences between TOMM on the one hand and the NAO robot used in [2], [3] on the other hand. First, the number of DOF increases from 5 DOF to 6 DOF. Second, the camera system on TOMM allows stereo tracking to obtain the distance of the tracked end-effector as third visual feature. Finally, the accessible workspace of TOMM is larger due to the increased size of the robot.

C. Developmental Process

Figure 2 shows the chain of phases during the development and the following gives a description for each of the steps.

1) Constrained DOF Exploration: After moving to a fixed home position, the robot starts to explore its own limb. Each DOF is moved within a short motion range of the total range starting from the home position for every available motion. Due to the small motion range constraint, this process is called constrained DOF exploration. The purpose of the exploration is to record and store multiple sequences which are later used for training. Each of the recorded sequences consists of a series of visual-proprioceptive patterns which contain the visual features of the tracked end-effector and the corresponding DOF states of each joint. Additionally to the available motions of each DOF, also one idle sequence is recorded where the robot does not move. Figure 3 shows a visualization of the constrained DOF exploration.

2) First-Stage Training: During this step, the PAS is trained to learn the correlations of DOF movement and the resulting visual feature changes. For this purpose, the recorded sequences from the constrained DOF exploration are used.

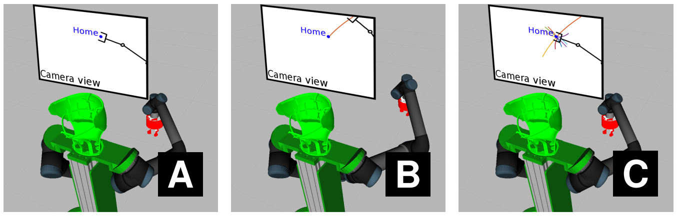

Fig. 3. Constrained DOF exploration starting from home position (A), then moving one DOF after the other (B) in both directions until sequences of each DOF are recorded (C).

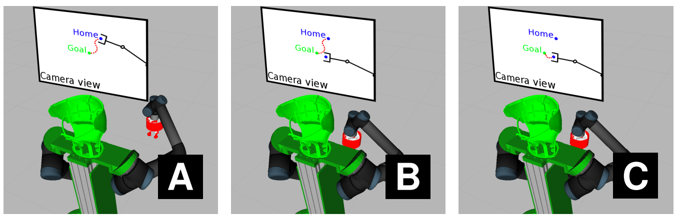

Fig. 4. Planning and execution of sequences during the reaching phase. (A) Internal mental simulation predicting sequence leading to received goal. (B) Sequence execution and prediction comparison. (C) Generation of new sequence after exceeding prediction error threshold until goal is reached or aborted.

3) First-Stage Reaching: The reaching phase is used to reach different goals in the robot's visual field by using the trained PAS in first-stage mode. Starting from the home position, a goal position is selected and sent to the PAS. The goals can be either selected in predefined patterns in the visual field or generated randomly. Mental simulation generates an imagined visual-proprioceptive sequence which brings the tracked end-effector closer to the goal. The observed visual-proprioceptive sequences during the reaching phase are again recorded for later use in the second-stage training. Figure 4 shows a visualization of the sequence planning and execution with observation of the prediction error.

4) Second-Stage Training: During the second-stage training, the PAS is trained with multiple sequences to combine them into more complex motions (see fig. 2 for source of training data).

5) Second-Stage Reaching: After the training, the PAS is able to recall the trained sequences and choose between them at each step in order to compose a longer trajectory consisting of parts of the trained sequences. The difference to the first-stage mode is the method how the sequence is generated in the PAS. Instead of using the relative changes in the visual and proprioceptive features for selecting the motion of each DOF independently, the PAS uses fragments of the absolute trained sequences to compose the imagined trajectory. All the observed sequences obtained during the second-stage reaching phase are again recorded and can be used for subsequent training. The second-stage PAS does not replace the first-stage PAS. It serves as a complement for increasing the knowledge of the overall system by generalizing across learned sequences.

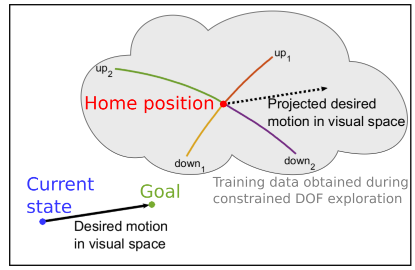

Fig. 5. Simplified example for two DOF: The figure shows the field of view of the robot during first-stage action selection. The colored lines represent the *up* and *down* motions of the two robot DOF learned during the constrained DOF exploration starting from the home position. The desired motion in visual space represents the delta between the current visual state and the desired visual state. The delta is projected to the home position and used to select the appropriate joint motions. In the given example the system selects an *up* motion for the first DOF and a *down* motion for the second DOF.

II. System Description

A. Mental Rehearsal and Mental Simulation

For action selection, a key capability is the prediction of changes which occur as consequences of actions and their real-world evaluation [15], [16]. The built-in MTRNN of the PAS recalls learned sequences. We define this recall of learned sequences as mental rehearsal. The recalled sequences are in turn used by the PAS algorithm to select the action. Here, we introduce the feedback of the (visual-proprioceptive) PAS output pattern to its input, instead of sending the proprioceptive pattern immediately to the robot. We refer to this closed-loop operation of PAS as mental simulation. In case a certain prediction error threshold is reached, the system generates a new sequence from the latest reached state.

B. Predictive Action Selector

1) MTRNN: In order to realize mental rehearsal, we use the MTRNN [17] modified to work with sigmoid neurons as explained in [3]. The MTRNN is trained using Backpropagation Through Time (BPTT) [18]. Mental rehearsal is realized by initializing the context states with the learned initial states and running the neural network for the desired number of iterations. Since the MTRNN utilizes sigmoid neurons, an interface layer to the PAS maps all inputs and outputs of the robot to the range between 0.0 and 1.0.

2) First-Stage Prediction: In the first-stage prediction, the PAS uses three separate motions for each DOF to compose complex motions: First, the *idle* motion which resembles no movement of the DOF at all, then an *up* motion where the DOF moves in the positive direction, and lastly a *down* motion in the negative direction. Algorithm 1 shows the process of generating sequences by first-stage prediction. In first-stage prediction, each DOF is analyzed separately to choose between the available motions. In order to select the best motion of a single DOF, the trained sequences resembling the three (k) available motions (*up*,*down*,*idle*) are recalled from the MTRNN. The resulting visual-proprioceptive sequences are then used to calculate the predicted relative feature changes (Δv_k, Δp_k). By comparing the predicted changes (Δv_k) to the

Algorithm 1 Generate sequence using first-stage prediction

input: visual goal v_g, init. visual end-effector state v_0 and joint state p_0
$[v, p] := [v_0, p_0]$
initialize empty generated sequence $seq_{gen} := []$
for i < prediction_length **do**
  $\Delta v := v_g - v$
  if $\Delta v <$ reached_threshold **then**
    break
  end if
  for each DOF **do**
    for motion k $\in [up, down, idle]$ **do**
      calculate $[\Delta v_k, \Delta p_k]$ by recalling from MTRNN
    end for
    select winning motion w by choosing k with $min(\Delta v - \Delta v_k)$
    $p := p + \Delta p_w$
    $v := v + \Delta v_w$
  end for
  $seq_{gen} := [seq_{gen}, [v, p]]$
end for
return seq_{gen}

Algorithm 2 Generate sequence using second-stage prediction

input: visual goal v_g, init. visual end-effector state v_0 and joint state p_0
$[v, p] := [v_0, p_0]$
initialize empty generated sequence $seq_{gen} := []$
for i < prediction_length **do**
  $\Delta v := v_g - v$
  if $\Delta v <$ reached_threshold **then**
    break
  end if
  for $k \in$ trained sequences **do**
    recognize context $[c_{FC}, c_{SC}]$ by recalling k from MTRNN
    initialize MTRNN context layers with c_{FC} and c_{SC}
    set MTRNN IO layer activations using $[v, p]$
    run MTRNN to generate prediction $[v_{pred}, p_{pred}]_k$
    compute predicted Euclidean distance to goal $\Delta v_k := |v_g - v_{pred}|$
  end for
  select winning sample w where $\Delta v_w == min\,(\Delta v_k)$
  $[v, p] := [v_{pred}, p_{pred}]_w$
  $seq_{gen} := [seq_{gen}, [v, p]]$
end for
return seq_{gen}

desired visual feature changes ($\Delta v = v_g - v$), the best suited motion for the current DOF is selected and the relative feature changes are applied to the current state. After repeating this motion selection for each DOF, the current state is appended as new sample to the generated sequence. The PAS cycles through this generation of samples until the set sequence prediction length is reached or until the generated sample is close enough to the goal. Figure 5 shows the simplified motion selection process of a 2 DOF robot. For a detailed description of the first-stage PAS algorithm we refer to [3].

3) Second-Stage Prediction: In contrast to the first-stage prediction mode, the second-stage prediction mode uses absolute trained sequences instead of computed relative changes. Figure 6 shows the generation of a sequence leading to the goal by using the five trained sequences which were, for example, recorded during a first-stage reaching phase. Furthermore, algorithm 2 shows the process of sequence generation in the second-stage mode. In second-stage prediction, each trained sequence is checked separately to select a new sample that is then appended to the generated sequence. For each sequence, the closest context in the MTRNN is found by recalling the

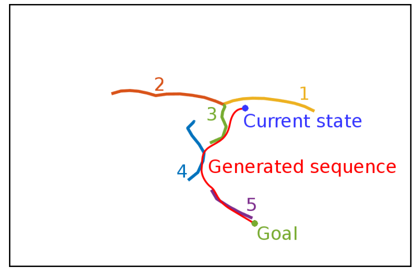

Fig. 6. The figure shows the field of view of the robot during second-stage sequence generation. The colored lines represent five different observed and learned sequences. By using the PAS in a closed-loop circuitry in second-stage mode for multiple steps, the red sequence is generated as an imagined trajectory leading to the goal.

sequence and comparing the predicted *Input/Output* activations to the current visual-proprioceptive sample. This process, which is called context recognition, initializes the activation of each context neuron in the neural network for the prediction. Then, for each trained sequence, the MTRNN is computed for one iteration to generate a predicted sample and the Euclidean distance of the visual features to the goal is computed. The predicted sample closest to the goal is chosen as winning sample and added to the generated sequence. For the next generation step, the predicted sample is set as new current visual-proprioceptive state. The PAS then cycles through this generation of samples until the set sequence prediction length is reached or until the generated sample is close enough to the goal.

III. Experimental Results

The following sections present the results of the experiments conducted on TOMM. First, the training data obtained from the constrained DOF exploration is shown. Then, the obtained training data is used for a first-stage reaching experiment of 378 goals in the visual field. Finally, the second-stage reaching, using training data obtained during first-stage reaching, is compared to the first-stage reaching for a test set of 27 goals. The selected goals are distributed using a grid pattern in the visual space. A color blob tracker is used to track the end-effector in both camera images to obtain the 3D visual feature vector which is given to the PAS. During the experiments a goal was considered reached when the Euclidean distance to the current state falls below 0.05 which corresponds to approximately 35mm in the center of the workspace. The maximum sequence prediction length for the mental simulation was set to 10 samples. Furthermore, the prediction error threshold used to regenerate a new predicted sequence was set to 0.15. We set a slow velocity for security reasons and to avoid problems with the industrial robot position control. As a result of this, the reaching times are much higher than the necessary computation time of the system.

A. Constrained DOF Exploration

Figure 7 shows the twelve motion sequences recorded during the constrained DOF exploration executed on TOMM.

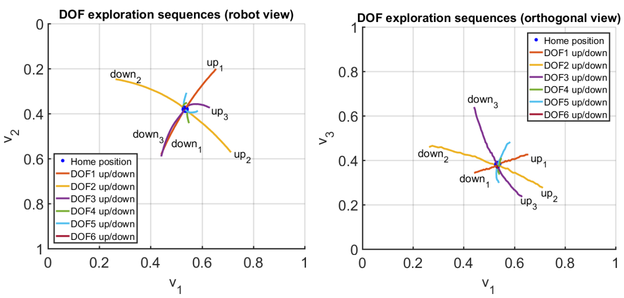

Fig. 7. Visualization of the sequences recorded during the constrained DOF exploration used as training data for the first-stage PAS. Plots show the sequences recorded during up and down motion of each joint in two different views. Left: v_1,v_2. Right: v_1,v_3. The motion direction indications are only added for three DOF to avoid clutter.

TABLE I
MTRNN PARAMETERS USED FOR THE EXPERIMENTS WITH 378 GOALS AND 27 GOALS.

Reaching	Goals	N_{IO}	N_{FC}	N_{SC}	τ_{IO}	τ_{FC}	τ_{SC}
1st-stage	378	9	10	10	7	20	20
1st-stage	27	9	10	10	5	10	100
2nd-stage	27	9	10	10	5	10	100

Each DOF was moved for 10% of the total available motion range in both directions which leads to an exploration of only 20% of the full joint range. Additionally, an idle sequence was recorded.

B. First-Stage Reachability

For the first-stage reaching the training set consisted of only 13 sequences with up to 29 samples each, which covered 20% of the motion range for each DOF around a selected home position. The constrained DOF exploration and the consecutive MTRNN training, with the parameters shown in table I, took only a total of 10 minutes. Part A of fig. 8 shows the resulting reachability map. From the 378 generated goals, a total of 66% were reached in the set reaching time limit of 180 seconds. Most of the missed goals are located farther away from the home position and closer to the borders of the images. Especially on the left side of the visual field, the tracking was often lost due to the simple design of the used color blob tracker. The average reaching time of the successfully reached goals was 72 seconds.

C. Development of Second-Stage Reaching

Following the first-stage reaching, the developmental process is continued by training the observed sequences to a second-stage PAS and using it for second-stage reaching. To validate this approach and the functionality of the second-stage PAS, this section presents the results and reachability maps from two different reaching experiments with a test set of 27 goals. Both experiments used identical settings except for the PAS mode (first-stage or second-stage) and thus different training data. The maximum goal reaching time was set to 180 seconds and the end-effector returned to the home position after each successful or canceled goal reaching attempt. Table I shows the parameters of the MTRNN used for the first-stage

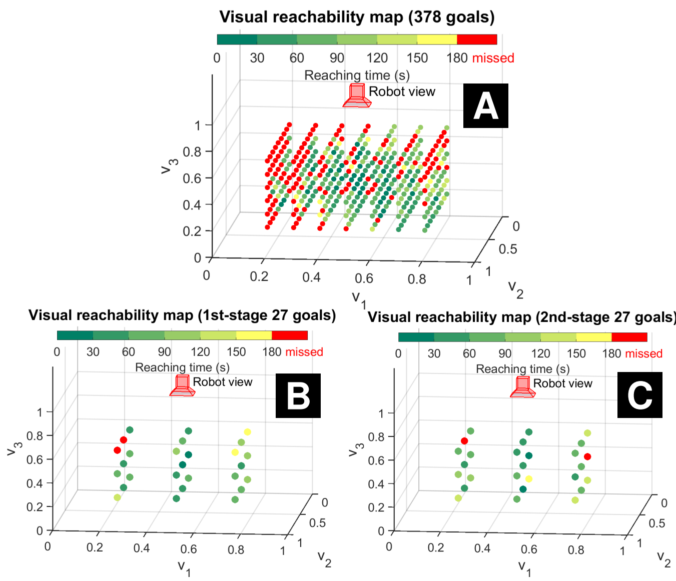

Fig. 8. Visual reachability maps obtained during first-stage reaching of 378 goals (A) as well as during the first-stage (B) and second-stage (C) reaching for a test set of 27 goals in the visual field. Each visual state goal is represented by a red dot (not reached) or a gradually colored dot depending on the reaching time. The first-stage PAS was trained with 13 sequences with a maximum of 29 samples each, obtained during the constrained DOF exploration. The second-stage PAS was trained with 57 sequences consisting of a total of 449 samples recorded during first-stage reaching. The axes of the plots represent the normalized three dimensional visual space.

PAS as well as for the second-stage PAS. When running in second-stage mode, the system falls back to first-stage mode with the first generated sequence after exceeding the reaching time of 60 seconds. As described in the last section, the first-stage PAS was again trained using the 13 sequences obtained during the constrained DOF exploration. The second-stage PAS on the other hand was trained using 57 sequences with a total of 449 samples observed and recorded during first-stage reaching of various goals.

The results of the two experiments are shown in the reachability maps (Fig. 8) and the reaching times (Table II). During both experiments, 92% of the target goals were reached after an average of approximately 77 seconds. On the one hand, the second-stage PAS was able to reach 9 goals before falling back to the first-stage mode ($t_{reach} < 60s$) which reduced their average reaching time by 21%. Furthermore, the reaching times of other goals were reduced by up to 59% using the second-stage PAS and the consecutive first-stage fallback and one goal was reached that was missed before in the first-stage reaching. On the other hand, the second-stage PAS seems to have insufficient knowledge to generate appropriate sequences for some of the goals. For 10 goals, the reaching time significantly increased ($> 15s$) and one goal, which was reached in pure first-stage mode, could not be reached anymore. Due to the simple fallback strategy, the reaching time increases when the second-stage predictor does not have sufficient knowledge to reach the goal. Since the maximum total reaching time for one goal is not increased, this can evidently also lead to missed goals which were reached

TABLE II

REACHING TIME RESULTS FOR SMALL TEST SET OF 27 GOALS EXECUTED WITH FIRST-STAGE PAS AND SECOND-STAGE PAS USING FALLBACK TO FIRST-STAGE MODE AFTER 60 SECONDS. THE SHORTEST REACHING TIME IS COLORED GREEN.

v_{g1}	v_{g2}	v_{g3}	1st stage (13 trained sequences)	2nd stage (57 trained sequences)
0.25	0.25	0.20	89.33	106.05
0.25	0.25	0.40	60.95	67.13
0.25	0.25	0.60	34.63	71.82
0.25	0.50	0.20	55.12	43.11
0.25	0.50	0.40	38.47	34.27
0.25	0.50	0.60	missed	missed
0.25	0.75	0.20	147.74	137.46
0.25	0.75	0.40	81.73	98.23
0.25	0.75	0.60	missed	68.27
0.50	0.25	0.20	63.79	152.17
0.50	0.25	0.40	13.50	14.67
0.50	0.25	0.60	59.48	58.65
0.50	0.50	0.20	40.86	29.90
0.50	0.50	0.40	11.62	43.12
0.50	0.50	0.60	64.96	32.08
0.50	0.75	0.20	56.74	88.16
0.50	0.75	0.40	51.44	80.75
0.50	0.75	0.60	104.66	59.85
0.75	0.25	0.20	92.73	134.86
0.75	0.25	0.40	104.76	missed
0.75	0.25	0.60	170.09	123.51
0.75	0.50	0.20	72.68	73.49
0.75	0.50	0.40	44.03	89.49
0.75	0.50	0.60	119.85	65.00
0.75	0.75	0.20	88.44	124.28
0.75	0.75	0.40	84.62	56.05
0.75	0.75	0.60	176.07	72.18
$mean(t_{reach})$			77.13	76.98
$N_{reached}/N_{goals}$			92.59%	92.59%

priorly. A possibility for improvement may include predictor switching, where the best suited PAS is selected depending on given goals.

IV. CONCLUSION

We presented a method for multi-stage developmental learning for reaching. The proposed system uses the Predictive Action Selector (PAS) by Wieser and Cheng [2], [3] as main building block for bootstrapping of goal-directed sequences. We extended the original system by introducing the second-stage mode and the mental simulation as two new features of the PAS. With the second-stage mode we gain a multi-stage developmental process with self-generation of training data for subsequent stages. After each stage, the knowledge of the system increases for improving the reaching performance. Including this work, our proposed system was evaluated on different robots with 2 DOF [2], 5 DOF [3] and now 6 DOF. Thus, we continued showing the DOF scalability and the general applicability of the system on different robotic platforms with altered kinematics. The validation on TOMM showed promising results for the proposed system. During the first-stage reaching of 378 goals in the visual field, a reachability rate of 66% was achieved using the 13 sequences recorded during the constrained DOF exploration. With a total of 349 visual-proprioceptive samples, these first training sequences covered only 20% of each DOF motion range around the initial home position. The second-stage reaching used a training set of 57 sequences containing 449 visual-proprioceptive samples obtained during first-stage reaching. Compared to the first-stage mode, the second-stage mode of the PAS was able to reduce the goal reaching times by up to 59%. In contrast to the first-stage PAS, the second-stage PAS can be continuously retrained with increasing training data in subsequent stages. Future work includes the further analysis of how the increase of training data influences the performance of the second-stage PAS.

REFERENCES

[1] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: A survey," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, May 2009.

[2] E. Wieser and G. Cheng, "Predictive action selector for generating meaningful robot behaviour from minimum amount of samples," in *2014 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Oct. 2014, pp. 139–145.

[3] ——, "Progressive learning of sensory-motor maps through spatiotemporal predictors," in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Sept. 2016, pp. 43–48.

[4] E. Dean-Leon, B. Pierce, F. Bergner, P. Mittendorfer, K. Ramirez-Amaro, W. Burger, and G. Cheng, "TOMM: Tactile Omnidirectional Mobile Manipulator," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2441–2447.

[5] R. Saegusa, G. Metta, G. Sandini, and S. Sakka, "Active motor babbling for sensorimotor learning," in *2008 IEEE International Conference on Robotics and Biomimetics*, Feb. 2009, pp. 794–799.

[6] Z. Mahoor, B. J. MacLennan, and A. C. McBride, "Neurally plausible motor babbling in robot reaching," in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Sept. 2016, pp. 9–14.

[7] N. Srinivasa and S. Grossberg, "A self-organizing neural model for fault-tolerant control of redundant robots," in *2007 International Joint Conference on Neural Networks*, Aug. 2007, pp. 483–488.

[8] P. Shaw, D. Lewkowicz, A. Giagkos, J. Law, S. Kumar, M. Lee, Q. Shen, and C. D. M. d'Autume, "Babybot challenge: Motor skills," in *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Aug. 2015, pp. 47–54.

[9] D. Luo, F. Hu, Y. Deng, W. Liu, and X. Wu, "An infant-inspired model for robot developing its reaching ability," in *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Sept. 2016, pp. 310–317.

[10] A. Dearden and Y. Demiris, "Learning forward models for robots," in *19th International Joint Conference on Artificial Intelligence (IJCAI 05)*, 2005, pp. 1440–1445.

[11] K. Narioka and J. J. Steil, "U-shaped motor development emerges from goal babbling with intrinsic motor noise," in *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, Aug. 2015, pp. 55–62.

[12] R. Rayyes and J. J. Steil, "Goal babbling with direction sampling for simultaneous exploration and learning of inverse kinematics of a humanoid robot," in *Proceedings of the workshop on New Challenges in Neural Computation*, 2016.

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

[14] C. von Hofsten, "Developmental changes in the organization of pre-reaching movements," *Developmental Psychology*, vol. 20, no. 3, pp. 378–388, 1984.

[15] R. C. Miall and D. M. Wolpert, "Forward models for physiological motor control," *Neural Networks*, vol. 9, no. 8, pp. 1265–1279, 1996.

[16] P. Haggard, "Conscious intention and motor cognition," *Trends in Cognitive Sciences*, vol. 9, no. 6, pp. 290–295, 2005.

[17] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment," *PLoS Computational Biology*, vol. 4, no. 11, p. e1000220, 2008.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Cognitive modeling, ch. Learning representations by back-propagating errors*, T. A. Polk and C. M. Seifert, Eds. MIT Press, Cambridge, 2002.