Learning Whole Upper Body Control with Dynamic Redundancy Resolution in Coupled Associative Radial Basis Function Networks

René Felix Reinhart and Jochen Jakob Steil

Abstract—We present a dynamical system approach to learning forward and inverse kinematics of a humanoid upper body in associative radial basis function networks. Coupling of arm kinematics via the torso joints is modeled by dynamically coupling two networks learning the direct inverse kinematics of both torso-arm chains separately. Dividing the upper body kinematics in two problems significantly reduces the number of samples required for learning. Redundancies of the inverse kinematics are represented by multi-stable dynamics of the associative networks and are resolved dynamically depending on the current system state. The model is exploited for task space tracking in a feedback control framework.

I. INTRODUCTION

The increasingly complex designs of robotic platforms, in particular regarding deformable materials and passive compliant actuators, challenge analytic techniques to obtain kinematic models for control. Learning of forward and inverse kinematics is therefore increasingly important. From a machine learning viewpoint, inverse kinematics pose two serious problems: (i) The representation of ambiguities due to redundancies of the kinematic chain, and (ii) the resolution of ambiguities during system exploitation, i.e. selecting a suitable solution out of many. Ambiguity of inverse relations is commonly tackled by multiple experts [1], [2] or probabilistic models [3]. Ambiguity resolution then requires additional circuitry during system exploitation and is not trivial. It is further difficult to a priori choose the number of experts for complex inverse mappings. Other approaches restrict the inverse model to a single solution which resolves the non-convexity problem of one-to-many relations for the learning. Ambiguity resolution then requires additional circuitry during system exploitation and is not trivial. It is further difficult to a priori choose the number of solutions, i.e. attractors, change over the input domain, which is achieved by the learning in a pure data-driven manner. That is, the number of solution branches must not be known in advance.

In this paper, we present an improved variant of the associative learning scheme, which utilizes a hidden layer of radial basis functions [9] instead of sigmoidal neurons as in [8]. Related to this approach are prototype-based versions of Echo State Networks [10] and Recurrent Self-organizing Maps [11]. In contrast to the recurrent connectivity of the hidden layers in these previous works, the associative radial basis function networks in this paper implement recurrent output feedback loops as in [8] but feature more efficient learning: Even though the model in [8] already restricts learning to a linear read-out layer of an otherwise randomly initialized and untrained hidden layer, this model requires the synthesis of data to imprint multi-stable dynamics. The associative radial basis function networks cope with multi-stable dynamics without additional data synthesis and thus render learning of the linear read-out layer very efficient.

The second contribution of this paper is the modular modeling of coupled kinematic chains which significantly mitigates the problem of exhaustively sampling joint configurations for the learning – a notoriously difficult problem for many degrees of freedoms that bothers all attempts to model complete inverse kinematics. For instance, consider the upper body of the humanoid robot iCub which comprises three torso joints coupling the kinematics of both arms (compare Fig. 1). Learning a complete inverse model without further knowledge about the kinematic structure requires the exhaustive sampling of the entire joint space. In case of controlling the first four joints of the upper arms and the torso of iCub, sampling the combined eleven-dimensional joint space.
space with six steps per joint yields to $6^{11} \approx 20,000,000$ samples. We approach this issue by training two networks, one for each torso-arm chain with seven degrees of freedom, such that the torso sub-chain is shared by both models. Then, the amount of training data reduces to $2 \cdot 6^7 \approx 600,000$ samples in this example. During system exploitation, both networks are dynamically coupled in a feedback control loop: The joint estimates for the torso of both models are superimposed and yield a combined feed-forward control command. Feedback of the actual torso joint angles into the network results in a dynamic, closed-loop negotiation of suitable torso postures which meet task constraints for both arms. We show that the dynamical system approach to learning kinematics is well-suited to implement the dynamical coupling of kinematic chain representations and yields to a flexible control architecture. The utility of the modular architecture is demonstrated in task space tracking scenarios under varying control conditions.

II. LEARNING FORWARD AND INVERSE KINEMATICS

We learn forward and inverse kinematics with an associative neural network model (see Fig. 2). We start with a notational convention to facilitate the further discussion. In the outset of associative methods, there is no dedicated input or output: With respect to the network architecture and the learning algorithm, task space variables $x$ and joint space variables $q$ are functionally equivalent (compare Fig. 2). For kinematics learning, it is nevertheless convenient to name $x$ as input and $q$ as output, because generally we drive the robot towards targets (inputs) in task space coordinates $x$.

The associative setup resolves the non-convexity problem arising from ambiguous data pairs $(x_n, q_n)$ and $(x_m, q_m)$ with $x_n = x_m$ and $q_n \neq q_m$ by utilizing a mixed representation $h(x_n, q_n) \neq h(x_m, q_m)$ of inputs and outputs in the hidden layer. Subscripts $n \neq m$ denote the index of data samples. Trained networks are applied in an output feedback-driven mode, i.e., estimated outputs are fed back into the network (compare Fig. 3). This feedback loop exhibits multistable attractor dynamics corresponding to the multiplicity of the solutions to the inverse kinematic problem [8]. The case of infinitely many solutions to the inverse kinematics corresponds to continuous attractor manifolds. Resolution of ambiguities proceeds dynamically by iterating the output feedback loop, i.e. the network settles to a particular attractor depending on initial conditions. This feed-forward control scheme of the direct kinematics using feedback of predicted joint angles is later on extended to a feedforward-feedback controller that receives actual joint angles from the robot. We first introduce the neural model and the output feedback-driven operation modes. Then, the network training procedures are presented.

A. Associative Radial Basis Function Networks (ARBF)

We consider the network architecture depicted in Fig. 2. The network comprises an input layer with neurons $x \in \mathbb{R}^D$ and $q \in \mathbb{R}^O$ as well as a hidden layer of radial basis functions $h(x, q) \in \mathbb{R}^N$ with centers $C_{\text{inp}} \in \mathbb{R}^{N \times D}$ and $C_{\text{out}} \in \mathbb{R}^{N \times O}$. The basis function activities for all neurons $i$ are computed by

$$h_i(x, q) = \frac{\exp(-a_i(x, q))}{\sum_{j=1}^N \exp(-a_j(x, q))}$$

$$a_i(x, q) = \frac{1}{D} \sum_{d=1}^D \frac{1}{\sigma_{\text{inp}}^d} (x_d - c_{\text{inp}}^d)^2 + \frac{1}{O} \sum_{d=1}^O \frac{1}{\sigma_{\text{out}}^d} (q_d - c_{\text{out}}^d)^2,$$

where $\sigma_{\text{inp}}^d$ and $\sigma_{\text{out}}^d$ are the radii (spreads) of the basis functions along each dimension $d$. Note that the weighting of quantization errors in (1) by their respective dimension yields a balanced contribution of these errors to the network state $h(x, q)$. This is important to render networks likewise responsive to changes of inputs or outputs [13].

The hidden layer is linearly read out to estimate outputs $\hat{x}$ and $\hat{q}$ according to connection weights $W_{\text{inp}}$ and $W_{\text{out}}$:

$$\hat{x}(x, q) = W_{\text{inp}} h(x, q)$$

$$\hat{q}(x, q) = W_{\text{out}} h(x, q).$$

B. Output Feedback-driven Exploitation

Association of task space coordinates and joint angles is accomplished by iterating the output feedback-driven network dynamics similar to Jordan-type recurrent neural networks [6] or to reservoir networks with output feedback [13]. To query the inverse kinematics, the network is driven by external inputs $x^*$ while estimated outputs $\hat{q}$ from the last iteration step are fed back into the hidden state in a recursive loop (compare Fig. 3). Equation (1) of the network dynamics changes to

$$a_i(x^*, q) = \frac{1}{D} \sum_{d=1}^D \frac{1}{\sigma_{\text{inp}}^d} (x^*_d - c_{\text{inp}}^d)^2 + \frac{1}{O} \sum_{d=1}^O \frac{1}{\sigma_{\text{out}}^d} (q_d - c_{\text{out}}^d)^2.$$
For the forward kinematic model, which is driven by $q^*$, equation (1) alters to
\[
a_i(\hat{x}, q^*) = \frac{1}{D} \sum_{d=1}^{D} \frac{1}{\sigma_d^{inp}} (\hat{x}_d - c_{id}^{inp})^2 + \frac{1}{D} \sum_{d=1}^{D} \frac{1}{\sigma_d^{out}} (q_d - c_{id}^{out})^2.
\]

The partial combination of both, the forward and inverse model, is also possible, which implements a mixed constraint satisfaction in task and joint space. Then, only parts of the task space and joint space inputs are externally driven while the remaining components are iterated in the output feedback loop (compare [8]).

C. Unsupervised Fitting of Basis Functions

In principle, all common algorithms for fitting basis functions to the data can be applied. We fit the basis functions in the combined task and joint space by an on-line k-means algorithm, which is a standard algorithm for this purpose and efficient to compute. The centers of the basis functions are updated for each data pair $(x, q)$ according to
\[
\Delta c_{m}^{inp} = \eta (x - c_{m}^{inp}),
\]
\[
\Delta c_{m}^{out} = \eta (q - c_{m}^{out}),
\]
where $m = \arg \min_{i \in \{1, \ldots, N\}} (a_i(x, q))$ is the index of the basis function center with the minimal distance to the current data point. The learning rate $\eta$ is set to 0.001 throughout all experiments and results do not critically depend on the chosen learning rate.

After fitting the centers, the spreads of the basis functions $\sigma_d^{inp}$ and $\sigma_d^{out}$ are set to the average pairwise distances between the centers $c_{id}^{inp}$ and $c_{id}^{out}$ for all hidden neurons $i = 1, \ldots, N$ such that we obtain ellipsoid basis functions. This heuristic yields well distributed responsibilities of the basis functions in the convex hull of the training data.

D. Supervised Read-out Learning

Supervised training of the read-out layer can be accomplished with common error minimization techniques. We apply linear regression to compute the read-out weights $W^{inp}$ and $W^{out}$ in one shot after fitting the basis functions. First, the hidden states $h$ are collected row-wise for each input-output data pair in a matrix $H$. Then, the read-out weights are calculated using linear regression:
\[
W^{inp} = (H^T H)^{-1} H^T X \tag{4}
\]
\[
W^{out} = (H^T H)^{-1} H^T Q \tag{5}
\]

To handle large data sets, we compute the matrix products $H^T H$ and $H^T X$ in mini-batches with 1000 samples each. In doing so, the required memory for the linear regression is constant with respect to the number of training samples.

III. LEARNING THE UPPER BODY KINEMATICS OF iCub

In this section, the data collection and training procedure for learning the kinematics of the humanoid robot iCub is outlined. We train two models, one model for each arm of iCub including the torso joints. In this paper, we restrict the models for simplicity to the three torso joints and the first four degrees of freedom of each arm and control the end effector position only. We denote the end effector positions for the left and right arm by $x_l$ and $x_r$. The same notation applies to the joint angles, where the first three components of $q_l$ and $q_r$ contain the three torso joints, respectively.

A. Sampling of Kinematic Data

Data is generated by sampling the full range of the seven-dimensional joint space for each arm (including the torso joints) according to an equally spaced grid with six steps per dimension yielding 279,936 data points per arm. We calculate the forward kinematics for each sample to obtain pairs $(x_l, q_l)$ and $(x_r, q_r)$ of task space positions and joint angle configurations. This step can be substituted by, for example, visually observing the end effector position. For testing, we sample targets $x_l$ and $x_r$ in task space on a three-dimensional grid comprising $5^3$ targets in a $40 \times 40 \times 35 cm$ cube around the home position of each arm. Note that the target points are not part of the training set but reside in the convex hull of the training data.

B. Network Training

We train associative networks with $N = 5,000$ basis functions for each arm. Larger hidden layer sizes will generally lead to more accurate fitting of the training data, whereas we observed that too small networks, e.g. with $N = 100$ basis functions, are not sufficient to regularly spread the responsibilities of basis functions in the entire input space. The centers of the basis functions are initially set to $N$ randomly chosen samples from the training set. Then, the k-means update rules (2)–(3) are applied for five sweeps through the entire data set with randomized presentation order. Finally, the read-out learning is conducted according to (4)–(5). We repeat the network training for each arm five times to access the performance variance over several network initializations.
C. Generalization of Trained Kinematics

We first evaluate the generalization performance to untrained target positions for the networks of each arm separately in a stand-alone condition. We compute the forward and inverse kinematic estimates for each arm and present errors in meters. In case of the inverse kinematics, e.g. of the left arm, the error for target \( x_l^* \) is calculated by

\[
E_l^{task}(x_l^*) = ||x_l^* - FK_l(\hat{IK}_l(x_l^*))||,
\]

where \( \hat{IK}_l \) and \( FK_l \) denote the learned inverse and actual forward kinematics, respectively.

The first two rows in Tab. I display test errors with standard deviations for the learned forward and inverse kinematics. The average error over all networks is in the millimeter range which, on the one hand, confirms the excellent interpolation abilities of radial basis function networks [9]. On the other hand, the accurate response to the task space inputs by the output feedback dynamics point out the utility of radial basis functions for associative computations: The small error variances in the first row show that training and exploitation via output feedback dynamics is robust over several network initializations. Tab. I confirms previous results for learning forward and inverse models in associative networks [8], [13]: The forward mapping is typically approximated more accurately than the inverse mapping which reflects the different difficulties of both mappings.

In the remainder of this paper we proceed with the best performing networks on the inverse kinematics for each arm. These two networks achieve very small absolute errors with small variance over all target positions (second row in Tab. I).

### IV. DYNAMICAL COUPLING OF ARM KINEMATICS

To control the upper body of iCub, we couple the networks for the left and right arm in a feedback control framework (see Fig. 4). The estimated torso joints of both networks are superimposed according to

\[
\hat{q}_t = \frac{1}{2} \left( (\hat{q}_{t1}, \hat{q}_{t2}, \hat{q}_{t3})^T + (q_{r1}, q_{r2}, q_{r3})^T \right)
\]

while the remaining arm joints are directly controlled by each network. The superimposed torso command is send to the robot. Instead of feeding back the estimated torso outputs of each network, the superimposed output is fed back such that both networks are informed of the actual robot joint configuration and adopt their inverse estimates in the next iteration if necessary. From the viewpoint of a single network, the coupling of network responses can be understood as external perturbation of the output feedback dynamics. We already showed in previous work that the external freezing of joint angles results in appropriate adaptation of the inverse estimate by the output feedback loop [8].

In the following experiments, we evaluate the performance of the coupled system using the best networks for the left and right arm from Section III.

### A. Dynamical Negotiation of Torso Postures

We evaluate the accuracy of the coupled model systematically on the grid of target positions from Section III in two different control conditions. In the first condition (Coupled), the right end effector is constrained to reside in the home position while the targets for the left arm are varied over the test grid. In the second condition (Right Unconstrained), only the left end effector is constrained to target positions on the test grid while the right end effector is not constrained (estimated positions \( \hat{x}_r \) are fed back into the right network).

Tab. I presents the averaged errors over all targets with standard deviations for both conditions in the last two rows. While the errors increase slightly for the coupled model in comparison to the stand-alone evaluation, the errors still reside below one centimeter on average. The slight increase of errors for the inverse kinematics of the left arm in the coupled condition points out the challenge of exploiting arm redundancies to meet the task constraints for mutually negotiated torso postures. Errors of the coupled system

![Fig. 4. Coupled feedback control framework: Target positions \( x_l^* \) and \( x_r^* \) (black) as well as the current joint angles \( q_l \) and \( q_r \) (gray) are fed into the Associative Radial Basis Function (ARBF) networks. Estimated joint angles \( \hat{q}_l \) and \( \hat{q}_r \) as well as estimated end effector position \( \hat{x}_l \) and \( \hat{x}_r \) are read out (black). The torso joint estimates of each model are superimposed.](image)
We present the flexible control of task and joint space constraints enabled by the associative representation of the robot’s kinematics. We constrain the right end effector to reside on a vertical line by externally driving the task space inputs of the network for the right arm only with $x_{r1}$ and $x_{r2}$. The remaining input $x_{r3}$ for the vertical end effector position is left unconstrained, which means that estimated values $\hat{x}_{r3}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

B. Multi-stable Dynamics Represent Redundant Solutions

We verify that multiple solutions to the inverse kinematics are stored in the networks and can be recalled depending on initial conditions. Fig. 5 (e)–(g) show three different elbow configurations for the left arm which are obtained by initially forcing the second joint of the upper arm to increasing values before the networks take over control. The network for the left arm memorizes the initial condition and stays in a rather lower elbow configuration (compare Fig. 5 (e)), a moderate elbow abduction (compare Fig. 5 (f)), or an upper elbow solution (compare Fig. 5 (g)). The robust exploitation of redundancies by the associative radial basis function networks confirms earlier results in [8] and demonstrates the capability to represent multiple solutions to the inverse kinematics by means of multi-stable output feedback dynamics.

C. Mixed Control in Task and Joint Space

We present the flexible control of task and joint space constraints enabled by the associative representation of the robot’s kinematics. We constrain the right end effector to decrease again in the second condition with unconstrained right end effector positions (last row in Tab. I). Then, the network for the right arm completely commits to the configuration proposed by the network for the left arm. That less constraints enable more effective task fulfillment is a general principle in control and applies also to the dynamically coupled controller of ARBF networks.

Fig. 5 shows exemplary postures obtained with the coupled control scheme. The dynamical coupling of both networks results in negotiated torso postures depending on the queried task space targets. For instance, targeting both end effectors to the left of the robot’s body, a twisted torso to the left brings both arms closer to the target area (compare Fig. 5 (a)). If, however, the right end effector is not constrained to a target in the left area of the robot, or is even left completely unconstrained, the coupled dynamics result in a moderate twist of the torso only (compare Fig. 5 (b)). The same effect is illustrated in Fig. 5 (c) and (d) for targets far in the front of the robot’s body: While for both targets the coupled dynamics result in an exaggerated forward bending of the torso, the torso remains rather upright if only a single target is far away from the robot’s body. In both cases, the targets are reached accurately (red spheres in palm).

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.

D. Tracking Task Space Trajectories

We finally present results for tracking a figure-eight pattern in task space using the simulation of iCub’s physical dynamics [12]. Tracking is conducted in a transient-based computation mode meaning that the output feedback dynamics are not iterated until convergence for each task space input. We present tracking results for three control conditions:

In the first condition, the right end effector is constrained to reside in the home position while the left end effector receives target inputs following the figure-eight pattern. The remaining input $x$ is left unconstrained, which means that estimated values $\hat{x}$ are fed back into the network. In addition, we constrain the first joint of the right arm $q_{r1}$ to take increasing values which corresponds to an increasing anteversion of this arm.

Fig. 6 (a)–(c) shows the resulting postures. The end effector complies with the constraint on the vertical line while increasing the angle of the first joint forces the selection of different solutions along the admissible set of end effector positions on the line. We have already shown in [8] that the networks comply with mixed constraints in task and joint space within the range of the overall accuracy of the trained models. Application of mixed task and joint space constraints emphasizes the flexibility of the associative network.
The average tracking error for the left and right end effector is 1.7 cm and 0.58 cm, respectively. Note that these errors include the temporal delay of the tracker response to the target pattern. Additionally to the tracking performance of the inverse estimates, the solid gray lines in Fig. 7 show the estimated positions of the end effectors $\hat{x}_l$ and $\hat{x}_r$. The end effector positions are also accurately estimated during the movement with average errors of 1.38 cm and 0.54 cm for the left and right end effector.

In the second condition, the network for the right end effector receives no task space inputs, i.e. estimated values $\hat{x}_r$ are fed back into the network. The unconstrained right end effector allows for a more flexible exploitation of the torso joints to achieve the target positions of the left end effector, which results in a slightly reduced tracking error of 1.51 cm (compare Fig. 7 (middle)). The right end effector now displays a drift mostly caused by the torso movements while its end effector position is estimated very accurately from the joint space feedback (see solid gray lines in Fig. 7 (middle)).

In the third condition, both end effectors track a figure-eight pattern simultaneously (compare Fig. 7 (right)). Also in this condition, the target pattern is tracked with high accuracy: The average tracking error of the inverse estimate in this condition, the target pattern is tracked with high accuracy: The average tracking error of the inverse estimate of 1 cm simultaneously (compare Fig. 7 (right)). Also in this condition, the target pattern is tracked with high accuracy: The average tracking error of the inverse estimate of 1 cm (compare Fig. 7middle)). The right end effector now displays a drift mostly caused by the torso movements while its end effector position is estimated very accurately from the joint space feedback (see solid gray lines in Fig. 7 (middle)).

Note that in all conditions two periods of the figure-eight patterns are tracked. Redundancies, however, are resolved equally in each pattern period: The networks remain in the current attractor regime of the output feedback dynamics and in this way achieve a cyclic resolution of ambiguities. The tracking results are excellent also in relation to other approaches to learning inverse kinematics [2], [5]. In contrast to the work in [2], [5], our model features the simultaneous estimation of forward kinematics and a flexible task space control interface, which enables arbitrary switching between constrained and unconstrained network components.

V. CONCLUSION

We presented a dynamical system approach to learning forward and inverse kinematics of a humanoid upper body. Deploying a decoupling strategy based on prior knowledge about the kinematic chain significantly mitigates the problem of data sampling for the learning. During system exploitation, torso configurations are dynamically negotiated between models through mutual feedback. The dynamical system approach to learning ambiguous inverse kinematics in associative networks is well-suited for this decoupling strategy, achieves excellent results while being efficient to train and providing a flexible control interface.

An extension point of the presented approach is to control the remaining three joint angles of each arm to achieve desired end effector orientations. This can be accomplished by dynamically coupling models for orientation control with the networks for end effector positioning.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community’s 7th Framework Program FP7/2007-2013, Challenge 2 - Cognitive Systems, Interaction, Robotics - under grant agreement 248311 - AMARSi.

REFERENCES