Unsupervised Body Scheme Learning through Self-Perception

Jürgen Sturm Christian Plagemann Wolfram Burgard

***Abstract*— In this paper, we present an approach allowing a robot to learn a generative model of its own physical body from scratch using self-perception with a single monocular camera. Our approach yields a compact Bayesian network for the robot's kinematic structure including the forward and inverse models relating action commands and body pose. We propose to simultaneously learn local action models for all pairs of perceivable body parts from data generated through random "motor babbling." From this repertoire of local models, we construct a Bayesian network for the full system using the pose prediction accuracy on a separate cross validation data set as the criterion for model selection. The resulting model can be used to predict the body pose when no perception is available and allows for gradient-based posture control. In experiments with real and simulated manipulator arms, we show that our system is able to quickly learn compact and accurate models and to robustly deal with noisy observations.**

I. Introduction

Kinematic models are widely used in robotics, in particular for prediction and control of robotic manipulators [1], [2]. Such models are typically derived analytically by an engineer [3] and usually rely heavily on prior knowledge about the robots' geometry and kinematic parameters. As robotic systems become more complex and versatile, however, or are delivered in a completely reconfigurable way, there is a growing demand for techniques allowing a robot to automatically learn body schemes with no or minimal human intervention. Such a capability would not only facilitate the deployment and calibration of new robotic systems but also allow for autonomous re-adaptation when the body scheme changes, e.g., through regular wear-and-tear over time or even intended reconfiguration in the case of tool use.

Neuro-physiological evidence indicates that humans as well as higher primates learn and adapt their internal models continuously and autonomously using self-perception [4]. Brain scan studies on monkeys that have been trained to use tools revealed that the tool itself even gets integrated into their body schemes over time [5]. Mirror neurons as found in brain area F5 map proprioceptive sensations to tactile and visual ones and thereby seem to serve as a neurological representation of the body scheme [6]. Moreover, they seem to translate external visual stimuli, for example from a demonstrator, into proprioceptive ones, and thereby play an important role in imitation and imitation learning.

In this paper, we investigate ways of realizing such capabilities on artificial systems, in particular on robotic manipulators in conjunction with visual self-perception. We

The authors are with the University of Freiburg, Department of Computer Science, Georges-Koehler-Allee 79, 79110 Freiburg, Germany
E-mail: {sturm, plagem, burgard} @ informatik.uni-freiburg.de

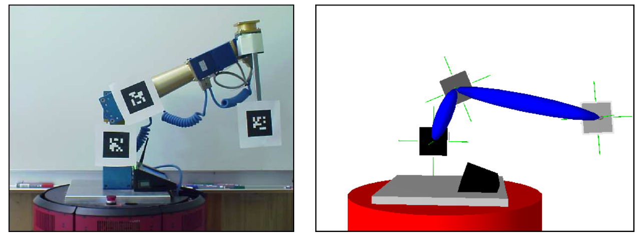


Fig. 1. Experimental setup: the robot issues random commands ("motor babbling") to its joints and perceives the resulting movements of its body parts using a monocular camera. From this self-perception, it learns a compact Bayesian network that it can then use both for prediction and control. The right picture shows a visualization of the robot's self-model after learning.

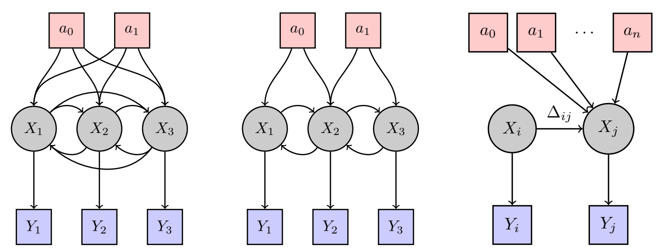


Fig. 2. **Left:** Initially, the Bayesian network representing the robot's body scheme is fully-connected. **Middle:** After training, only the local models most consistent with the observed data are retained to form a sparse kinematic model for the whole system. **Right:** Template of a local model for a body part X_j which depends on its predecessor X_i in the kinematic chain and all available action commands $a_1, \ldots, a_n$.

propose to learn a Bayesian network for the robot's kinematic structure including the forward and inverse models relating action commands and body pose. More precisely, we start with a fully connected network containing all perceivable body parts and available action signals, to perform random "motor babbling," and to iteratively reduce the network complexity by analyzing the perceived body motion. At the same time, we learn non-parametric regression models for all dependencies in the network, which can later be used to predict the body pose when no perception is available or to allow for gradient-based posture control. In experiments with real and simulated manipulator arms, we show that our approach is able to quickly learn compact and accurate models and to robustly deal with noisy observations.

II. Related Work

Several approaches for learning and adapting body schemes at different levels of complexity have been proposed in the past. Self-calibration [7], for instance, can be understood as a subproblem of body scheme learning. When the kinematic model is known up to a number of parameters, they can in certain cases be efficiently estimated by maximizing the likelihood of the model given the data. Genetic algorithms have been used in [8] for parameter optimization when no closed form is available. To a certain extend, such methods can also be used [9] to calibrate a robot that is temporarily using a tool. However, such approaches require a parameterized kinematic model of the robot.

There have also been approaches on learning sensor-motor maps when no such model is available. [10] used for example Hebbian networks to discover the body scheme from self-occlusion or self-touching sensations and later [11] learned classifiers for body/non-body discrimination from visual data. Other approaches used for example nearest-neighbor interpolation [12] or neural networks [13]. By considering body scheme learning as a problem of function approximation, such approaches are applicable in cases even where little prior knowledge is available. Without assuming any underlying structure, these approaches however generalize badly over the training data and therefore scale badly with an increasing number of free variables.

This problem can be tackled by reducing the dimensionality of the learning problem. Principal component analysis (PCA) for example has been used successful [14] for walking gait learning on humanoid robots. Although such approaches remove efficiently the redundancy in the body scheme for a particular motion sequence, much information is lost in the projection as the low-dimensional mapping only describes a reduced body scheme. Another possibility for dimension reduction is by unveiling the underlying structure of the body scheme. In [15], this is formulated as a model selection problem between different Bayesian networks. Here, the qualitative relation between actions and observations is learned that describes the observed data well. By using this structural information, the robot can infer motor commands by which it imitates the movements of a human demonstrator. To our knowledge, in this approach however only the structure was learned without quantitative relationships such that so far no precise actuation has been realized.

In contrast to all of these approaches, we propose an algorithm that both learns the structure as well as an accurate functional mapping. By first selecting a suitable decomposition of the body scheme, local models with smaller complexity can be learned.

III. A Probabilistic Model for Kinematic Chains

The problem we are trying to tackle in this work is to enable a robotic system to autonomously learn the relationship between available action signals $a_1, \ldots, a_n$ and body part configurations $X_1, \ldots, X_m$, which can be (partially) observed as $Y_1, \ldots, Y_m$. In our concrete scenario, in which we learn the kinematic model of a robotic manipulator arm, the action signals a_i are real-valued variables corresponding to the individual states of the joints and the $X_i \in \mathbb{R}^{4\times 4}$ are homogeneous transformation matrices, each encoding the 6-dimensional pose of a body part relative to a reference coordinate frame. On the real robotic platform used in our experiments, the observations Y_i are obtained by tracking visual markers in 3D space including their 3D orientation [16]. Note that these observations are inherently noisy, especially in the z dimension, which is the distance of the marker from the camera, and we also consider markers that are only partially observable (e.g., just x and y) or cannot be detected at all.

We model the whole system as a Bayesian network, in which the body parts are arranged in a chain such that each X_j can be described by a local model $p(X_j|X_i, a_1, \ldots, a_n)$ given its (unique) predecessor X_i and the action commands as depicted in Fig. 2, as well as an observation model $p(Y_j|X_j)$. We denote the local transformation from X_i to X_j by $\Delta_{ij} = X_i^{-1} X_j$. Given no additional prior knowledge about the relationships between actions and body parts, learning in this model means

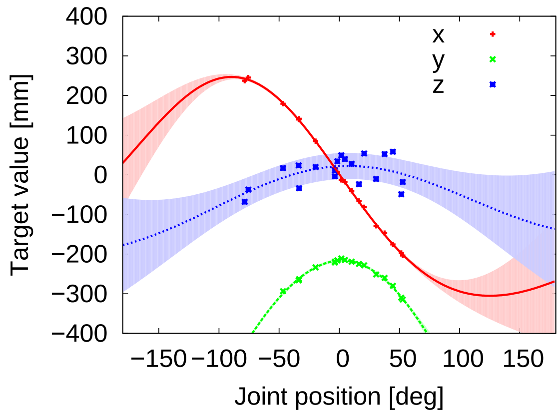
1) finding the correct network topology (which parts are directly connected?) and
2) learning the local transformation models for this topology

by issuing action commands $a_1, \ldots, a_n$ and observing the outcomes $Y_1, \ldots, Y_m$. In most applications, the local transformations Δ_{ij} depend on few action signals only and not on all n as in the general case. Thus, an important practical aspect for learning will be to select appropriate subsets of action commands for the individual local transformations.

A. Finding the Network Topology

We are looking for a compact Bayesian network for $p(X_1, \ldots, X_m|a_1, \ldots, a_n)$ that is composed of local models of the form $p(\Delta_{ij}|\mathbf{A}_{ij})$ with $\mathbf{A}_{ij} \subset \{a_1, \ldots, a_n\}$. Please note that a trivial solution would be to choose all local models to have full rank, that is, to simply depend on all action signals available. Such a model, however, would generalize badly over the training data as it would not take advantage of the intrinsic redundancy to the body structure. Since the individual models would then be of high dimensionality, consequently such a model would require considerably more training examples than a sparse composition of low-dimensional local models. The upper arm of a robot, for example, only depends on the position of the trunk and the shoulder joints, while the lower arm would only depend on the position of the elbow and the remaining joints. This of course needs not always to be the case: in the experimental section, we will also evaluate our system for the case, in which individual body parts are not observable and, thus, higher order local models have to be learned to be able to build a full model.

Considering all possible model dimensionalities and dependencies, the decomposition results in a search problem with an upper bound of $\sum_{k=1}^{n} \binom{n}{2}\binom{m}{k}$ local models that would have to be learned, i.e., as the ordering of joints and observed body parts is initially unknown to the robot. In practice, this number can be reduced drastically by using simple search heuristics, such as evaluating the local models ordered by their complexity $|\mathbf{A}_{ij}|$ and to interrupt the search when a certain level of model accuracy is attained.



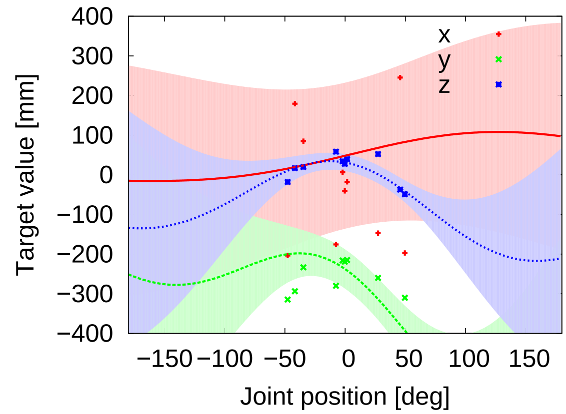


Fig. 3. **Left:** Example of an accurate local model learned for two body parts and an action variable. Note the low predictive variance for the x- and y components as well as the higher noise in the z dimension, which is due to higher measurement uncertainty in this direction. **Right:** Less accurate model learned for the same body parts but a different action variable. Such a local model is less likely to be part of the Bayesian network describing the full kinematic chain of the robot since, in general, its predictions are less accurate.

As the quality measure for hypothesized network topologies in the model selection process, we use cross-validation. More precisely, we determine for each local model $p(\Delta_{ij}|\mathbf{A}_{ij})$ the residual sum of squares (RSS) on a validation set that has been sampled during training but has not been included in the training set for the local models.

The Bayesian network can then be composed of a subset of the evaluated local models that minimizes the overall sum of RSS'. This subset can be found efficiently, e.g., by using a minimal spanning tree algorithm. Note, that model selection based on the RSS measure worked particularly well in our experiments, but other selection criteria like the Bayesian information criterion (BIC) can be used likewise.

As a result, the recovered Bayesian network factorizes the body scheme into the more compact representation

$$\begin{aligned} p(X_1,\ldots,X_m|a_1,\ldots,a_n) &= P(X_t) \prod_{<i,j>\in E} p(X_i|X_j,\mathbf{A}_{ij}) \\ &= P(X_t) \prod_{<i,j>\in E} p(\Delta_{ij}|\mathbf{A}_{ij}), \end{aligned} \tag{1}$$

where X_t is the root node and E is the edge list of the recovered minimal spanning tree corresponding to the kinematic chain(s).

B. Learning Local Kinematic Models

In order to learn an arbitrary local model $p(\Delta_{ij}|\mathbf{A}_{ij})$, we need to find the non-linear mapping from a vector of action signals $\mathbf{A}_{ij}$ to an expected relative transformation Δ_{ij}. For simplicity, we assume all 12 free components δ^k_{ij} of Δ_{ij} being independent of each other and thus consider the functional mapping for each component separately. In theory, this approach cannot guarantee that the result is a valid, homogeneous transformation matrix, e.g., the rotational components are orthogonal and positive-definite. It is a hard problem in general to find the best parameterization for constrained matrices in regression settings [17]. Since we did not observe any problem related to this issue in our experiments, however, we kept this formulation and will consider deriving a different parameterization in future research.

As the true relative transformations $\Delta_{ij} = X_i^{-1}X_j$ are only observable through the noisy observations $\overline{\Delta}_{ij} = Y_i^{-1}Y_j$, we assume additive white noise on each component $\overline{\delta}^k_{ij} \sim \mathcal{N}(\delta^k_{ij}, \sigma_n)$. A flexible model for learning such non-linear functions directly from noisy observations are the popular Gaussian processes. Due to space constraints, we only give the main characteristics of this framework here and refer to [18] for details. The main feature of the Gaussian process framework is, that the observed data points are explicitly included in the model and, thus, no parametric form of f needs to be specified. Moreover, the dependencies between data points is specified in an interpretable way using a parameterized covariance function k and predictions yield not only the most likely function value but also the corresponding predictive uncertainty. We parameterize the covariance function k using the often used square exponential formulation

$$k(\delta_{ij}, \delta_{rs}) = \sigma_f^2 \cdot \exp\left(-\frac{1}{2\ell^2}|\delta_{ij} - \delta_{rs}|\right) , \tag{2}$$

which depends on the Euclidian distance between points $\mathbf{x}_p$ and $\mathbf{x}_q$ as well as on the amplitude parameter σ_f^2 and the lengthscale ℓ. This covariance function is particularly well suited to model sinusoidal dependencies as they arise in our setting, where we wish to infer components of harmonic transformations.

The middle and right diagrams of Fig. 3 depict several typical regression results, both generated using real data of our manipulator. In the middle diagram, the regressions of an accurate local model are shown. It can be seen that the training data has low noise in the x-y-components and somewhat higher noise in the z-component. In the right diagram, the data is poorly correlated and, in comparison, will yield a much higher prediction error (RSS) on the

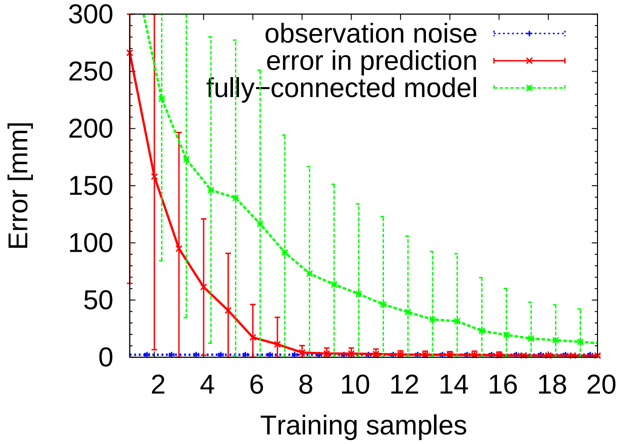


Fig. 4. Forward model evaluation on a simulated robot with low noise. The prediction error of the learned forward model quickly converges towards zero. It can also been seen that the compact Bayesian network – using a decomposition into small local models – converges much faster than the fully-connected Bayesian model which requires higher-dimensional regression models.

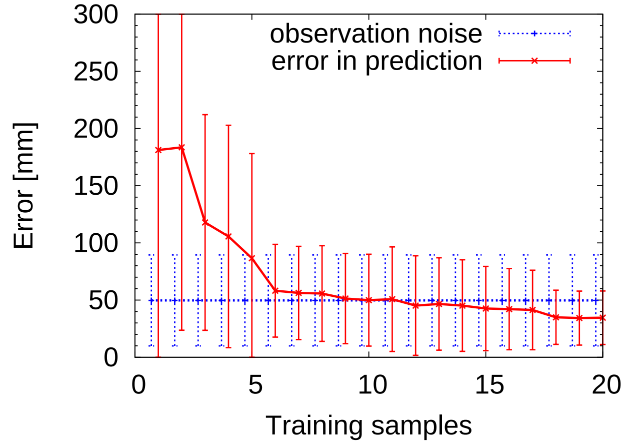


Fig. 5. Forward model evaluation on a real robot with noisy perception. Already after a few samples, the prediction accuracy of the forward model becomes better than the direct but noisy perceptions of the camera.

validation set. Therefore, the latter model is less likely to be part of the kinematic chain for the full system.

C. Using the kinematic chain for prediction

The learned Bayesian network can now be used as a predictive forward model. Given a motor command $a_1, \ldots, a_n$, the relative transformations Δ_{ij} can be inferred from the local models $p(\Delta_{ij}|\mathbf{A}_{ij})$ of the kinematic chain.

If one absolute body position, e.g., X_1 is known additionally, the absolute coordinates of all other body positions can be computed by re-arranging equation 1.

In particular, the GPs underlying each local model yield the mean and the variance for a given motor command. While the mean corresponds to the maximum likelihood estimate, the variance can be used as a measure of uncertainty. In order to propagate Gaussians beliefs through the kinematic chain, we approximate the result of Gaussian multiplication again as Gaussians [19].

Note that this variance estimates can be used by the robot for active exploration, or to generate action commands that minimize the expected sensor and/or motor noise.

D. Using the kinematic chain for control

In order to grasp an object, or to imitate the posture of a human demonstrator, the robot needs an inverse model that maps from a given target position X_{target} to a action command $\mathbf{a} = [a_1, \ldots, a_n]^T$ that is supposed to generate this position.

Depending on the complexity of the configuration space, different search algorithms can be used. When gradients are available, potential field approaches have been proven to be flexible and powerful solutions for maneuvering and path planning [2].

In our case, this translates to the distance function $f(\mathbf{a}) = \|X_m(\mathbf{a}) - X_{target}\|$ that has to be minimized. X_m refers here to the predicted position of body part m given a motor command $\mathbf{a}$, which can be computed from equation 1 as described in the previous subsection. As this distance function is continuous, also its Jacobian $\nabla f(\mathbf{a})$ can be evaluated, i.e.,

$$\nabla f(\mathbf{a}) = \left[\frac{\partial f(\mathbf{a})}{\partial a_1}, \ldots, \frac{\partial f(\mathbf{a})}{\partial a_n}\right]^T \quad (3)$$

A gradient descent algorithm can then be used to minimize $f(\mathbf{a})$ and thereby iteratively approach the target position.

IV. Evaluation

We have tested our approach in a series of experiments, both on a simulated manipulator robot and a real one. The experiments described in this section have been designed to demonstrate that

1) our approach yields a close to optimal and compact model when no noise is present and all quantities can be fully observed,
2) our approach is robust w.r.t. the noisy perception of a monocular camera on a real robot,
3) our approach can deal with unobserved body parts (in which case higher-order local models are needed),
4) our approach allows for free and real online control, when no perception is available.

For each experiment, 400 random action commands were generated ("motor babbling") and sent to the motors. After each action request was completed, the robot recorded the measurements from the joint encoders $a_1, \ldots, a_n$ and the observed positions $Y_1, \ldots, Y_m$ of its body parts.

These datasets were then used for learning, testing and validation. The training samples were added incrementally to the local models, in order to investigate the learning behavior. After each training sample, a test set of 40 data samples was used to measure the average accuracy of both prediction (forward model) and control (inverse model).

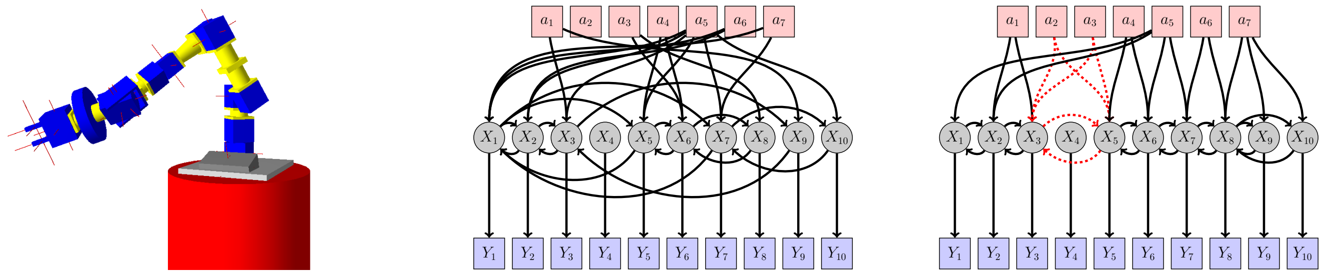


Fig. 6. Experiments with a simulated 7-DOF-manipulator consisting of 10 body parts. Body part X_4 was hidden and therefore never observed. **Left:** Screenshot from the simulated robot. **Middle:** Bayesian network after the first training sample: the correct kinematic chain can not yet be recovered. **Right:** Bayesian network after 10 training samples, the kinematic structure has converged to the true solution. Note that as only model of complexity one, i.e., of type $p(X_i|X_j, a_k)$, have been evaluated, the chain cannot be closed. Here, the robot would need to consider additionally models of complexity two, i.e. $p(X_i|X_j, a_k, a_l)$ and finally find a good model using $p(X_5|X_3, a_2, a_3)$ (indicated by the dashed red arrows).

A. Fully observable

For our first experiment, we used a simulated manipulator robot with 2 rotational joints, similar to our real robot as shown in Figure 1. We added small amounts of white noise to both the measurements from the joint encoders ($\sigma_{joints} = 0.02°$) and the observations of the body positions ($\sigma_{markers} = 1mm$). From the simulator, noise-free ground truth information was available for evaluation.

Fig. 4 shows the prediction errors of the learned model as a function of the number of training samples. Remember that a single training sample here corresponds to a pair of proprioceptive and visual observations $\langle a_1, \ldots, a_n, X_1, \ldots, X_m \rangle$. It can be seen that the prediction error quickly converges towards zero; after only 10 training samples, the error is in the magnitude of millimeters. For comparison, a full model $p(\Delta_{13}|a_1, a_2)$ was learned directly from the training data using a 2-dimensional GP. The resulting prediction error is also given in Fig. 4. It can be seen that the compact Bayesian network composed of two local models converges much faster than the fully-connected model.

B. Fully observable, with noise

The robot used to carry out the experiments is equipped with a manipulator composed of Amtec (Schunk) PowerCube modules. With nominal noise values of ($\sigma_{joints} = 0.02°$), the reported joint positions of the encoders were considered to be sufficiently accurate to compute the ground truth positions of the body parts from the known geometrical properties of the robot. Visual perception was obtained by using a Sony DFW-SX900 FireWire-camera at a resolution of 1280x960 pixels. On top of the robot's joints, black-and-white markers were attached (see Fig. 1), that were detectable by the ARToolkit vision module [16]. Per image, the system perceives the unfiltered 6-dimensional poses of all detected markers.

The standard deviation of the camera noise was measured to $\sigma_{markers} = 44mm$ in 3D space, which is acceptable considering that the camera was located two meters apart from robot. In the near future, we plan to develop a body part tracker similar to [20] that uses more natural visual features, such that the 6D trajectories can be recovered directly from images without the need for artificial marker tags.

Experiments with marker detection alone showed good accuracy in the xy-plane and rotations around the camera direction. However, the z-distance (distance from the camera) was more noisy, as well as the rotational estimates when the marker was turned away from the camera direction. In order to keep the observation noise low, it was decided to restrict the robot's movements physically to a plane perpendicular to the camera by only using 2 of its 4 joints. No further post-processing was applied, i.e., the robot did not know that it was physically restricted to a plane.

The measured noise levels were considerably higher (around 44mm). Still, the body scheme converged within the first 10 observations. After about 15 training samples, the accuracy of the predicted body part positions even outperformed the accuracy of the direct observations. The latter is a remarkable result as it means that, although all local models are learned from noisy observations, the resulting model is able to blindly predict positions that are more accurate than its direct perception.

C. Partially observable body parts

We conducted an experiment with a simulated manipulator with 7 joints and 10 visible body parts, with a total length of $1300mm$. The manipulator has been assembled as follows (compare to Fig. 6):

- Body parts X_1 and X_2 were firmly connected to each other.
- Two fingers X_9 and X_{10} were mounted on the 1-DOF gripper a_7 as the end-effector.
- The remaining body was constituted of a chain of visible body parts $X_2, \ldots, X_8$ and 1D rotary joints $a_1, \ldots, a_6$.

The learned forward model converges after only 10 samples, similar as in the earlier experiments. The average prediction error is then lower than $1mm$.

We then analyzed the effects of partial observability on our approach. This was realized by covering body part X_4 completely, such that no observations of that body part could be made. As a result, all models under consideration have a RSS larger than a certain threshold and therefore no suitable local model of complexity 1 can be found.

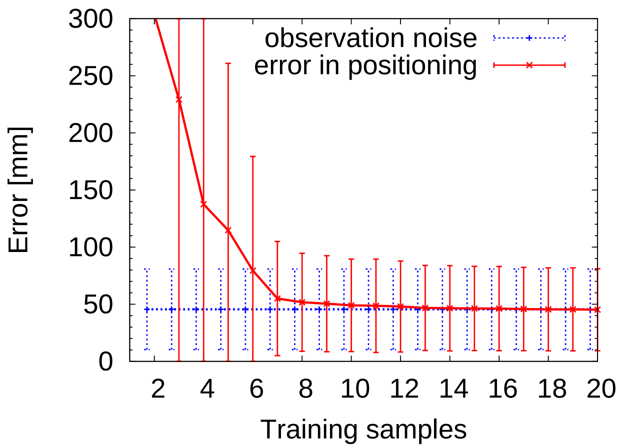


Fig. 7. Inverse model evaluation on a real robot with noisy perception. A gradient-descent algorithm is used to find the action command that minimizes the predicted distance to the desired target position. The positioning error reaches after a few samples the magnitude of the sensor noise.

In such a situation, the robot considers all local models of the complexity 2, etc., until a satisfactory kinematic chain is found, i.e., the predictive error over the test set (RSS) drops below a certain threshold. The higher order model which was automatically incorporated into the Bayesian network is highlighted in Fig. 6 by dashed red lines.

D. Manipulator control without perception

Finally, we evaluated the kinematic chain in inverse direction. Fig. 7 shows the results on the real robot with noisy perception. The average positioning error converges after 10 training samples approximately at the level of the observation noise. This result is slightly worse than the prediction accuracy of the forward model. A possible reason for this is, that here the robot has to deal with the observation noise twice: first, the model was learned from noisy data, and second, for evaluation the desired target position was supplied to the robot again from real and therefore noisy perception.

V. Conclusions

In this paper, we presented an approach that allows an autonomous robot to learn its own sensorimotor model through self-perception. Our fundamental idea is to decompose the problem of learning a large and complex kinematic model into smaller pieces. As a result, the robot no longer needs to rely on a model supplied by an engineer. As such a model can continously be learned and adapted by the robot, it can be easily kept up-to-date.

Despite our encouraging results, the work described in this paper implies several interesting directions for future research. During tool-use, for instance, the Bayesian network could temporarily be extended by an additional node. The robot would then only need to learn a new local model, describing the transition from its end-effector to the tip of the tool. Additionally, the kinematic structure in form of a predictive Bayesian network can be used to identify the geometrical structure of the robot. For example, it should be possible with basic geometry to recover the position of the rotary axes of the robot. A trajectory planner could then implement obstacle avoidance on the whole body.

Acknowledgment

This work has partly been supported by the EU under FP6-004250-CoSy and by the German Ministry for Education and Research (BMBF) through the DESIRE project.

References

[1] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, June 2005.

[3] E. M. Rosales and Q. Gan, "Forward and inverse kinematics models for a 5-dof pioneer 2 robot arm," University of Essex, UK, Tech. Rep. CSM-413, 2004.

[4] A. N. Meltzoff and M. K. Moore, "Explaining facial imitation: A theoretical model," *Early Development and Parenting*, vol. 6, pp. 179–192, 1997.

[5] A. Maravita and A. Iriki, "Tools for the body (schema)," *Trends in Cognitive Sciences*, vol. 8, no. 2, pp. 79–86, 2004.

[6] N. P. Holmes and C. Spence, "The body schema and the multisensory representation(s) of peripersonal space," *Cognitive processes*, vol. 5, no. 2, pp. 94–105, 2004.

[7] N. Roy and S. Thrun, "Online self-calibration for mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1999.

[8] J. C. Bongard, V. Zykov, and H. Lipson, "Automated synthesis of body schema using multiple sensor modalities," in *Proc. of the International Conference on the Simulation and Synthesis of Living Systems (ALIFEX)*, 2006, pp. 220–226.

[9] C. Nabeshima, M. Lungarella, and Y. Kuniyoshi, "Timing-based model of body schema adaptation and its role in perception and tool use: A robot case study," in *Proc. of the IEEE International Conference on Development and Learning (ICDL2005)*, 2005, pp. 7–12.

[10] Y. Yoshikawa, K. Hosoda, and M. Asada, "Binding tactile and visual sensations via unique association by cross-anchoring between double-touching and self-occlusion," in *Proc. of the International Workshop on Epigenetic Robotics*, Genoa, Italy, 2004, pp. 135–138.

[11] Y. Yoshikawa, Y. Tsuji, K. Hosoda, and M. Asada, "Is it my body? – body extraction from uninterpreted sensory data based on the invariance of multiple sensory attributes –," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2004, pp. 2325–2330.

[12] P. Morasso and V. Sanguineti, "Self-organizing body-schema for motor planning," *Journal of Motor Behavior*, vol. 26, pp. 131–148, 1995.

[13] L. Natale, "Linking action to perception in a humanoid robot: A developmental approach to grasping," Ph.D. dissertation, Univerity of Genoa, Italy, May 2004.

[14] D. Grimes, R. Chalodhorn, and R. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Proc. of the Robotics: Science and Systems (RSS)*, Philadelphia, USA, August 2006.

[15] A. Dearden and Y. Demiris, "Learning forward models for robots," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, 2005, pp. 1440–1445.

[16] M. Fiala, "Artag revision 1, a fiducial marker system using digital techniques," National Research Council Canada, Tech. Rep. NRC/ERB-1117, 2004.

[17] D. J. C. MacKay and R. Takeuchi, "Interpolation models with multiple hyperparameters," *Statistics and Computing*, vol. 8, pp. 15–23, 1998.

[18] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[19] R. Ware and F. Lad, "Approximating the distribution for sums of products of normal variables," University of Canterbury, England, Tech. Rep. UCDMS 2003/15, 2003.

[20] J. Yan and M. Pollefeys, "Automatic kinematic chain building from feature trajectories of articulated objects," in *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 712–719.