Body Schema Learning

Jürgen Sturm, Christian Plagemann, and Wolfram Burgard

Abstract This chapter describes how the kinematic models of a manipulation robot can be learned, calibrated, monitored and adapted automatically using the perception and actuation capabilities provided by the robot's middleware. The presented technology requires only minimal human intervention by building on the concepts of self-observation and non-parametric learning. Specifically, the approach is to learn the kinematic model of a robotic manipulator from scratch using self-observation via a single monocular camera. We introduce a flexible model based on Bayesian networks that allows a robot to simultaneously identify its kinematic structure and to learn the geometrical relationships between its body parts as a function of the joint angles. Further, we show how the robot can monitor the prediction quality of its internal kinematic model and how to adapt it when its body changes—for example due to failure, repair, or material fatigue. This chapter includes experiments carried out both on real and simulated robotic manipulators designed to verify the validity of the approach for real-world problems, such as end-effector pose prediction and end-effector pose control.

1 Introduction

Kinematic models are widely used in robotics to describe the mechanism of a robot. For example, the kinematic model of a manipulation robot is typically specified by the position of its joints, and the size and orientation of its links [Craig, 1989, Sciavicco and Siciliano, 2000]. Kinematic models are usually derived analytically by a robot engineer and thus rely heavily on prior knowledge about the geometry of the robot. When such a model is applied to a real robot, its parameters have to be carefully calibrated [Gatla et al., 2007] to ensure a high accuracy, for example, using expensive calibration systems at the robot manufacturer's site. As robotic systems become more versatile and are increasingly delivered in completely reconfigurable ways, there is a growing demand for techniques to learn kinematic models automatically. Ideally, such techniques would neither require human intervention nor costly calibration equipment. This capability does not only facilitate the deployment and calibration of new robotic systems but also enables robots to autonomously adapt their models when the kinematics change, for example, as a result of hardware failures or material fatigue. Furthermore, the intelligent use of tools also requires the robot to include a tool dynamically in its kinematic model [Nabeshima et al., 2006].

Jürgen Sturm
University of Freiburg, Dept. of Computer Science, Georges Köhler Allee 79, 79110 Freiburg, Germany,e-mail: sturm@informatik.uni-freiburg.de

Christian Plagemann
Stanford Artificial Intelligence Lab, 353 Serra Mall, Gates Building 1A, Stanford, CA 94305-9010, U.S.A e-mail: plagemann@stanford.edu

Wolfram Burgard
University of Freiburg, Dept. of Computer Science, Georges Köhler Allee 79, 79110 Freiburg, Germany,e-mail: burgard@informatik.uni-freiburg.de

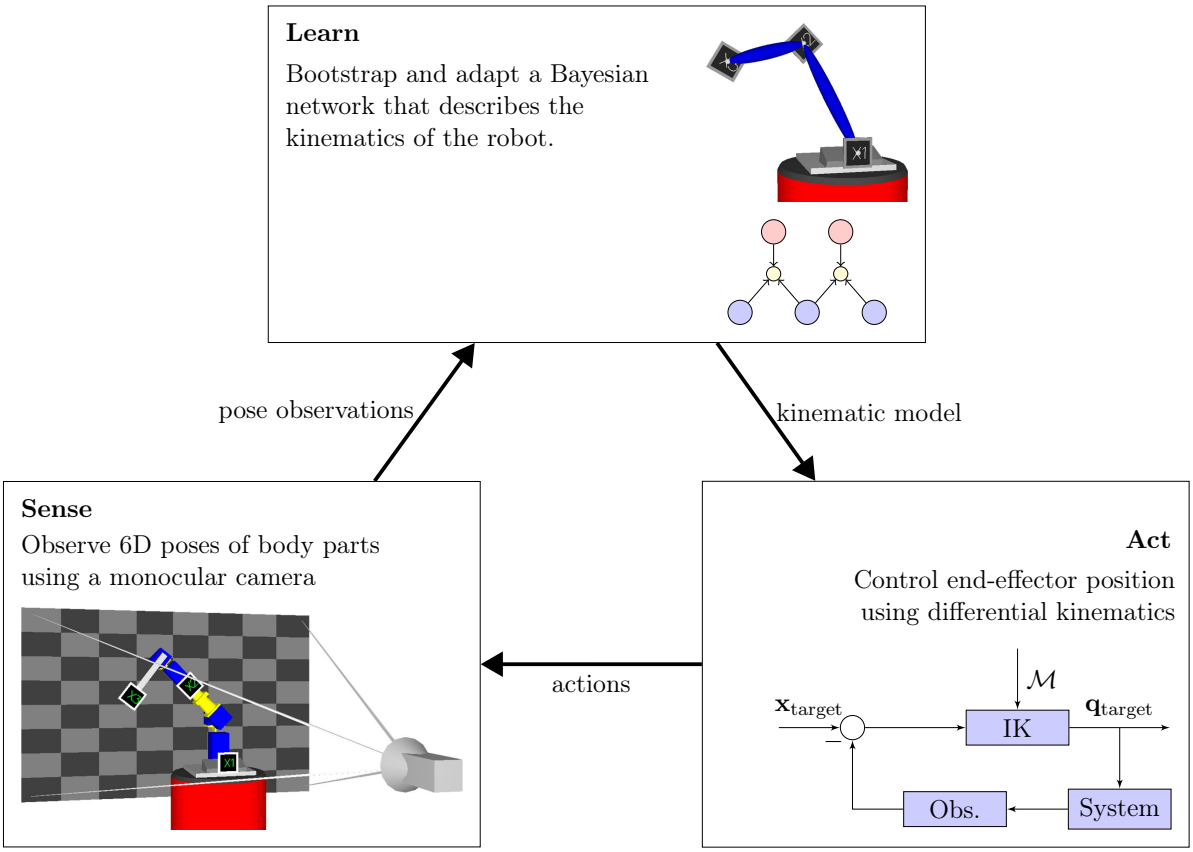


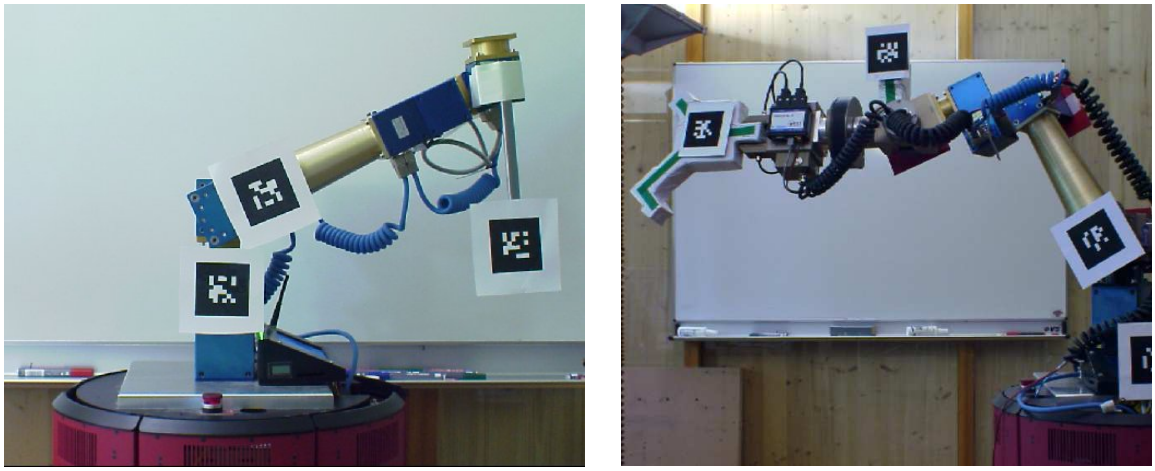
Fig. 1: Schematic overview of our approach to body schema learning.

The concept of kinematic models in robotics is closely related to the concept of the *body schema* in cognitive neuroscience [Stamenov, 2005, Gallagher, 2005] that refers to our internal representation of the body. Neuro-physiological experiments indicate that humans as well as higher primates adapt their body schema continuously [Meltzoff and Moore, 1997], for example, when handling tools [Maravita and Iriki, 2004]. In particular in the field of developmental robotics, various researchers have applied these concepts to robotic systems [Natale, 2004, Metta et al., 2006].

In this chapter, we develop a novel approach that allows a robot to learn its body schema using visual self-observation and exploratory actions. Our model is based on Bayesian networks that we use to represent the kinematic structure. We learn models for the individual joints of a robot using Gaussian process regression and develop an efficient algorithm to estimate the full kinematic structure of the robot. In experiments carried out in simulation and on real robots, we demonstrate that our approach enables a manipulation robot to learn its kinematic model from scratch and to maintain it over extended periods of time. Furthermore, we show that a robot using our approach can accurately predict and control the pose of its end effector even in the presence of hardware failures.

Figure 1 illustrates the proposed approach. The robot sends random "motor babbling" commands to its joints, observes the resulting pose, and estimates the kinematic model of itself from this sequence of observations. In each iteration, the robot learns Gaussian process models for the individual joints and searches for the kinematic structure that best explains the observed motion. The robot can use the learned model to predict and control the pose of its end effector. We developed and tested our approach on several simulated and two real manipulation robots as depicted in Figure 2.

This chapter is structured as follows. In Section 2, we briefly introduce kinematic models for manipulation robots and explain how they can be represented using Bayesian networks. Subsequently in Section 3, we present our probabilistic framework for learning such kinematic models from visual self-observations. In Section 4, we extend our framework to enable a robot to localize errors in the model and efficiently replace mismatching parts. In Section 5, we present experimental results obtained with



(a) 2-DOF robot (b) 7-DOF robot

Fig. 2: The manipulation robots used in this chapter to develop and test our approach.

real and simulated manipulator arms. These experiments demonstrate that our approach is able to learn compact and accurate models and is capable of dealing robustly with noisy observations. Finally, we conclude this chapter with a discussion of related work in Section 6.

2 Kinematic Models for Manipulation Robots

The kinematic model of a manipulation robot describes the relationship between its configuration and its body posture, i.e., the relationship between the joint angles and the poses of the body parts in 3D space. Figure 3a shows an example of a simple 2-DOF manipulation robot. The robot consists of two rotary joints q_1 and q_2, and five body parts $\mathbf{x}_1, \ldots, \mathbf{x}_5$. The first two body parts are connected rigidly. This means that the geometric transformation $\mathbf{\Delta}_{12}$ from the trunk $\mathbf{x}_1$ to the shoulder $\mathbf{x}_2$ is independent of the configuration of the joints. The shoulder $\mathbf{x}_2$ and the upper arm $\mathbf{x}_3$ are connected by the shoulder joint q_1, and thus their geometric transformation $\mathbf{\Delta}_{23}(q_1)$ depends on the joint angle of q_1. The same holds for the following parts, as the joint angle of the elbow joint q_2 has direct influence on the geometrical transformation $\mathbf{\Delta}_{34}(q_2)$ between the upper arm $\mathbf{x}_3$ and the lower arm $\mathbf{x}_4$. The gripper $\mathbf{x}_5$ is attached rigidly to the lower arm $\mathbf{x}_4$, such that $\mathbf{\Delta}_{45}$ is a fixed transformation. The kinematic function of this manipulator can thus be constructed by the concatenation of these individual transforms, i.e.,

$$f(q_1, q_2) := \mathbf{\Delta}_{12} \circ \mathbf{\Delta}_{23}(q_1) \circ \mathbf{\Delta}_{34}(q_2) \circ \mathbf{\Delta}_{45}. \tag{1}$$

The kinematic function $f(q_1, q_2)$ describes the full geometrical transformation from the coordinate frame of the trunk to the coordinate frame of the gripper. In engineering, the kinematic function of a manipulation robot is often constructed of the individual transformations by the specification of the *Denavit-Hartenberg (DH) parameters* [Sciavicco and Siciliano, 2000].

For many robotic applications, it is necessary to compute the configuration q_1 and q_2 to reach a given target position in the workspace. This requires the inversion of f, which is also called the inverse kinematic function. As the algebraic inversion is only possible for simple manipulators, a solution to the inverse kinematic problem is in practice often computed using an iterative numerical method such as the Jacobian transpose, pseudo-inverse or damped-least squares method [Buss and Kim, 2005].

A fundamental insight in our work is that the kinematic model of a manipulation robot can be represented in form of Bayesian networks. Consider the example given in Figure 3b: the configuration variables q_1 and q_2, the poses of the body parts $\mathbf{x}_1, \ldots, \mathbf{x}_5$, and the relative transformations $\mathbf{\Delta}_{12}, \ldots, \mathbf{\Delta}_{45}$ of our example robot appear as nodes in the Bayesian network. Further, the topology of the network encodes the kinematic structure: the relative transformation $\mathbf{\Delta}_{12}$ relates the first two body parts $\mathbf{x}_1$

and $\mathbf{x}_2$, while the second relative transformation $\mathbf{\Delta}_{23}$ depends additionally on the configuration q_1 of the first joint.

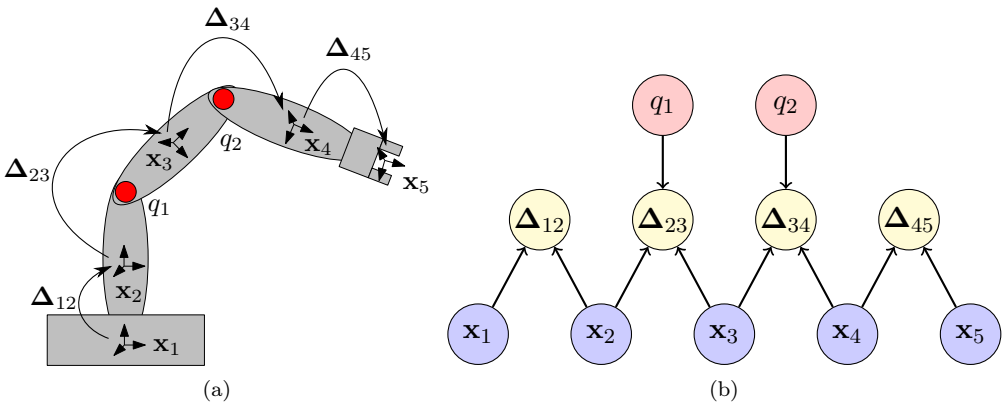


Fig. 3: (a) Simple 2-DOF manipulator consisting of 5 body parts. (b) The kinematic model of this robot represented as a Bayesian network.

We can now use standard inference techniques for Bayesian networks to predict the pose of the end effector (given $q_1, \ldots, q_m$ and $\mathbf{x}_1$, infer $\mathbf{x}_n$) or to control the pose of the end effector (given $\mathbf{x}_1$ and $\mathbf{x}_n$, infer $q_1, \ldots, q_m$). Both problems can be solved by marginalizing over all other variables in the network: solving forward kinematics corresponds to a marginalization over all intermediate body parts. As we will elaborate in the next section, this marginalization can be solved efficiently and in closed form when we assume that all variables in the Bayesian network are normally distributed.

3 A Bayesian Framework for Body Schema Learning

We define the robotic body schema as the joint probability distribution over joint actions $\mathbf{q} = (q_1, \ldots, q_m)$, true poses $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, and pose observations $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)$ of a manipulation robot. The individual $q_i \in \mathbb{R}$ are real-valued variables corresponding to the latest configuration request sent to the i-th joint of the robot. The $\mathbf{x}_i \in SE(3)$ encode the true poses of the body parts with respect to a reference coordinate frame. The $\mathbf{y}_i \in SE(3)$ are the robot's pose observations of its body parts that are generally noisy and potentially missing. Here, $SE(3)$ refers to the special Euclidean group that represents all three-dimensional poses (including both position and orientation). Internally, we represent these 3D poses as homogeneous $\mathbb{R}^{4\times4}$ matrices, which can be concatenated and inverted. We denote a sequence of t action-pose observations as $\mathcal{D} = \big\langle (\mathbf{q}^1, \mathbf{y}^1), (\mathbf{q}^2, \mathbf{y}^2), \ldots, (\mathbf{q}^t, \mathbf{y}^t) \big\rangle$. Formally, we seek to learn the probability distribution

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}_1, \ldots, \mathbf{y}_n \mid q_1, \ldots, q_m)\,, \tag{2}$$

which in this form is intractable for all but the simplest scenarios. Therefore, we assume that each observation variable $\mathbf{y}_i$ is independent from all other variables given the true pose $\mathbf{x}_i$ of the corresponding body part and that they can thus be fully characterized by an observation model $p(\mathbf{y}_i \mid \mathbf{x}_i)$. Furthermore, if the kinematic structure of the robot was known, a large number of pair-wise independencies between action signals and body parts could be assumed, which in turn would lead to the much simpler, factorized model

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid q_1, \ldots, q_m) = \prod_i p(\mathbf{x}_i \mid \text{parents}(\mathbf{x}_i))\,. \tag{3}$$

Here, parents($\mathbf{x}_i$) refers to the parent nodes of $\mathbf{x}_i$ in the Bayesian network and comprises only those body parts and action signals on which $\mathbf{x}_i$ directly depends on. Note that the actions are given and, thus, do not depend on other variables in this model. We now make the factorized structure of the

problem explicit by introducing hidden variables $\mathbf{\Delta}_{ij} := \mathbf{x}_i^{-1}\mathbf{x}_j$ corresponding to the relative geometric transformation between all pairs $(\mathbf{x}_i, \mathbf{x}_j)$ of body parts. Further, we denote with $\mathbf{z}_{ij} := \mathbf{y}_i^{-1}\mathbf{y}_j$ the relative geometric transformation relating the observations $\mathbf{y}_i$ and $\mathbf{y}_j$ that correspond to $\mathbf{x}_i$ and $\mathbf{x}_j$. Using this, we define as a *local model* the subgraph of our network that describes the geometric relationship between any two body parts $\mathbf{x}_i$ and $\mathbf{x}_j$ given the relevant part of the action signal, if all other body parts are ignored. Figure 4 shows a prototypical local model. Here, we denote with $\mathcal{Q}_{ij}$ the set of action signals that have a direct influence on $\mathbf{\Delta}_{ij}$. Any set of $(n-1)$ local models which forms a spanning tree over all n body parts defines a model for the whole kinematic structure and is a solution to Eq. (3).

Note that our approach does not require a proprioceptive sensor telling the robot in which configuration a particular joint is after executing an action q_i. At first sight, it seems that with proprioception one could learn the kinematic function passively from visual and proprioceptive observations only. While this is true, one would lack the mapping from motor commands to motor encoders such that the learned model would not suffice for manipulator control. One would either need to assume that motors and proprioceptive sensors are calibrated precisely, or one would need to additionally learn the mapping from actions to joint encoder values for each joint. In contrast to this, we learn a combined model that directly maps from motor commands to body pose observations. In this way, our approach closes the action-perception-loop, as visualized in Figure 1, and it obviates the need for the explicit calibration of the motor encoders.

In the following, we explain how to learn local models from data and how to find the spanning tree built from these local models that best explains the whole robot. We consider the single best solution only and do not perform model averaging over possible alternative structures. Note that in theory, it would be straight-forward to keep multiple structure hypotheses and to average over them for prediction using Bayes' rule. Control under structure uncertainty, however, is a slightly more difficult problem. One would have to consider all possible structures and assess the individual risks and gains for alternative actions. Then, one would select the action that maximize the overall gain while keeping all possible risks low. In practice, we found that considering the most-likely structure only is sufficient for most of the relevant tasks. Our approach is conservative in this respect since it requires a certain minimal accuracy from all parts of the body schema before the model is considered complete.

3.1 Local Models

The local kinematic models are the central concept in our body schema framework. A *local model* $\mathcal{M}$ (see Figure 4) describes the geometric relationship between two body parts i and j given a set of action signals $\mathcal{Q}_{ij}$. We propose to learn this relationship from data samples acquired while requesting random joint configurations and observing their effects on the robot's pose. As the learning framework for solving this supervised regression problem, we apply Gaussian process models for regression [Rasmussen and Williams, 2006]. The observations $\mathbf{y}_i$ of part locations $\mathbf{x}_i$ are obtained by tracking visual markers in 3D space including their position and orientation [Fiala, 2005]. These markers are also depicted in Figure 2. Note that the observations $\mathbf{y}_i$'s are inherently noisy and that missing observations are common, for example, in consequence of (self-)occlusion. Formally, the task is to learn the local transformations $\mathbf{\Delta}_{ij}$, each linking two body parts $\mathbf{x}_i$ and $\mathbf{x}_j$. Considering Figure 4, a straight-forward approach would be to infer the true poses $\mathbf{x}_i$ and $\mathbf{x}_j$ from the noisy observations $\mathbf{y}_i$ and $\mathbf{y}_j$, by assuming Gaussian white noise on the observations, i.e.,

$$\mathbf{y}_i \sim \mathcal{N}(\mathbf{x}_i, \Sigma_{\mathbf{y}}). \tag{4}$$

Then, one would need to integrate over the latent true poses $\mathbf{x}_i$ and $\mathbf{x}_j$ in order to reason about $\mathbf{\Delta}_{ij}$.

However, since the *absolute* positions $\mathbf{x}_i$ are irrelevant for describing the *relative* transformations, we take a slightly different approach by focusing directly on the transformations $\mathbf{z}_{ij}$ between observations $\mathbf{y}_i$ and $\mathbf{y}_j$. Note that these virtual measurements $\mathbf{z}_{ij}$ are noisy observations of the true transformation Δ_{ij} as a result of Eq. (4), i.e., we obtain

$$\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{\Delta}_{ij}, \Sigma_{\mathbf{z}}). \tag{5}$$

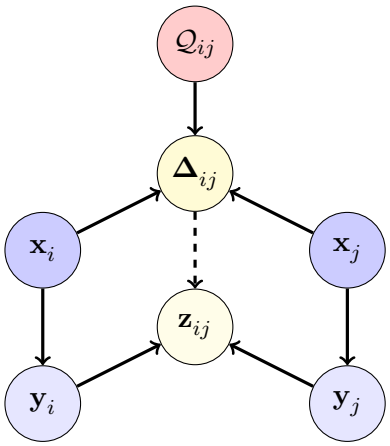


Fig. 4: Template of a local model that defines the kinematics between two related body parts.

With this, we can directly learn the relationships of actions $\mathcal{Q}_{ij}$ to relative transformations $p(\mathbf{z}_{ij} \mid \mathcal{Q}_{ij})$. The problem of learning a single local model now has the form of the noisy regression problem

$$\mathbf{z}_{ij} = f_{\mathcal{M}}(\mathcal{Q}_{ij}) + \boldsymbol{\epsilon} \tag{6}$$

that is, the regression function

$$\begin{aligned} f_{\mathcal{M}}: \quad & \mathbb{R}^{|\mathcal{Q}_{ij}|} \to \mathbb{R}^{16}, \\ & \mathcal{Q}_{ij} \mapsto \Delta_{ij} \end{aligned} \tag{7}$$

has to be learned from a sequence of noisy observations $\mathbf{z}_{ij}$.

For simplicity, we consider the over-parametrized transformation matrices in the following with $d = 12$ independent components and keep the remaining 4 elements of the homogeneous matrices fixed to (0 0 0 1). Subsequently, we learn the functional mapping for each of the 12 components separately. Due to this simplification, we cannot guarantee that all predictions correspond to valid, homogeneous transformation matrices. In practice, however, they lie close to valid transformations such that a normalization step resolves the problem. In particular, we ortho-normalize the rotational part of the homogeneous matrix using singular value decomposition. For solving the regression problem as stated in Eq. (7), we learn a Gaussian process model [Rasmussen and Williams, 2006] for the transformation functions $f_{\mathcal{M}}$ for all local models $\mathcal{M}$ and choose the squared exponential covariance function to parametrize the process.

An example of this is given in Figure 5. The red, green and blue curves show the translational x-, y-, and z- components of two different local models, respectively. The depicted models were learned from real data using Gaussian process regression. In the situation shown in Figure 5a, the action (x-axis) physically corresponds to the transformation being measured (y-axis). Thus, the data set is self-consistent and accurate functions with low noise levels can be learned. The higher noise level for the z-component is due to larger measurement error in this direction (i.e., the camera's line of vision). In the situation depicted in Figure 5b, a local model has been learned for variables that do not have a direct physical relationship. As a result, the model predicts the observations with a high uncertainty and thus does not explain the data well. Such a local model is likely to be discarded during the search for the full body model.

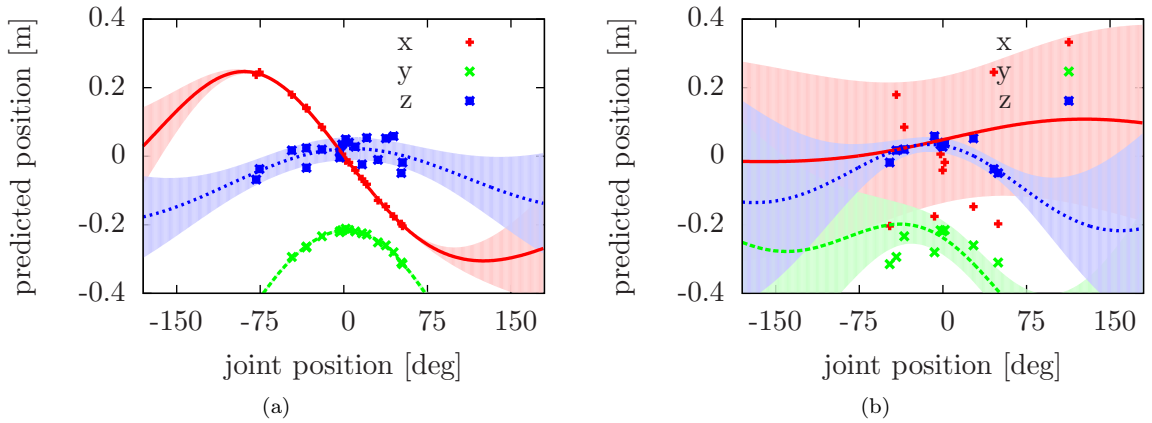


Fig. 5: Two local models learned from real data. (a) Example of an accurate local model. (b) Another local model that is less likely to be selected. The shaded areas represent the uncertainty of the learned Gaussian process.

3.2 Learning a Factorized Full Body Model

We seek to find a factorized model of the kinematic model that best explains the observed data. Our aim is to learn and evaluate this model efficiently, i.e., we aim to minimize the number of local models that need to be learned.

We implement this by discarding all local models that are overly inconsistent with the observed data. We define a local model $\mathcal{M}$ to be valid given a set of observations $\mathcal{D}$, if and only if the sample observation log-likelihood is above some threshold η, i.e.,

$$\frac{1}{|\mathcal{D}|} \log p(\mathcal{D} \mid \mathcal{M}) > \eta \tag{8}$$

that we will denote with the Boolean predicate $valid_{\mathcal{M}}(\mathcal{D})$. In practice, we use the 3σ confidence interval based on the sensor noise as a threshold to reject models that are overly inconsistent with the observations. We compute the data likelihood of a set of observations $\mathcal{D}$ as the product of the likelihoods of the individual observations, i.e.,

$$p(\mathcal{D} \mid \mathcal{M}) := \prod_{(\mathbf{z}_{ij}, \mathcal{Q}_{ij}) \in \mathcal{D}} p(\mathbf{z}_{ij} \mid \mathcal{Q}_{ij}, \mathcal{M}). \tag{9}$$

According to our observation model from Eq. (5), we assume Gaussian noise in the observations $\mathbf{z}_{ij}$ with covariance $\Sigma_{\mathbf{y}}$ with respect to the expected pose $\hat{\Delta}_{ij} := \mathbb{E}[\Delta_{ij} \mid \mathcal{Q}_{ij}, \mathcal{M}]$ as predicted from the Gaussian process model, resulting in

$$p(\mathbf{z}_{ij} \mid \mathcal{Q}_{ij}, \mathcal{M}) := \frac{1}{\sqrt{(2\pi)^6 \, |\Sigma_{\mathbf{y}}|}} \exp\left(-\frac{1}{2} (\mathbf{z}_{ij} - \hat{\Delta}_{ij})^T \Sigma_{\mathbf{y}}^{-1} (\mathbf{z}_{ij} - \hat{\Delta}_{ij}) \right). \tag{10}$$

To compare models with different data likelihoods and complexities, we define a *model quality measure* as

$$q(\mathcal{M}) := \underbrace{\log p(\mathcal{D} \mid \mathcal{M})}_{\text{accuracy}} - \underbrace{k \log(\eta \, |\mathcal{D}|)}_{\text{complexity}} \tag{11}$$

where $k \in \mathbb{N}$ denotes the dimensionality of the model $\mathcal{M}$, i.e., the number $|\mathcal{Q}_{ij}|$ of action signals that the model depends on. This measure is proportional to both the model accuracy and to a penalty term for model complexity. Note that this quality measure is similar to the Bayesian information criterion [Schwarz, 1978]. The key difference of our quality measure is that it contains the likelihood threshold as an additional factor in the complexity penalty. This formulation provides us two important

properties that we can exploit to specify an efficient search strategy for the kinematic structure. These two properties are:

- Given two models of the same complexity but different data likelihoods, the quality measure favors the model with the better data fit.
- Given two valid models with different complexity, the quality measure favors the model with the lower complexity.

The first property follows directly from the definition of the quality measure. The second property results from the definition of valid models in Eq. (8) in combination with the threshold as a factor in the model quality measure. If $k_1 < k_2$ and both models are valid, i.e., both $\log p(\mathcal{D} \mid \mathcal{M}_1) > \eta|\mathcal{D}|$ and $\log p(\mathcal{D} \mid \mathcal{M}_2) > \eta|\mathcal{D}|$, we can show that $q(\mathcal{M}_1) > q(\mathcal{M}_2)$ as follows:

$$\begin{aligned} q(\mathcal{M}_1) - q(\mathcal{M}_2) &= \log p(\mathcal{D} \mid \mathcal{M}_1) - k_1 \log(\eta\,|\mathcal{D}|) - [\log p(\mathcal{D} \mid \mathcal{M}_2) - k_2 \log(\eta\,|\mathcal{D}|)] \\ &> \log \eta\,|\mathcal{D}| - k_1 \log(\eta\,|\mathcal{D}|) - [\log p(\mathcal{D} \mid \mathcal{M}_2) - k_2 \log(\eta\,|\mathcal{D}|)] \\ &\geq \log \eta\,|\mathcal{D}| - k_1 \log(\eta\,|\mathcal{D}|) - [\log 1 - k_2 \log(\eta\,|\mathcal{D}|)] \\ &\geq \log \eta\,|\mathcal{D}| - k_1 \log(\eta\,|\mathcal{D}|) - [\log 1 - (k_1+1) \log(\eta\,|\mathcal{D}|)] \\ &= \log \eta\,|\mathcal{D}| - k_1 \log(\eta\,|\mathcal{D}|) - 0 + k_1 \log(\eta\,|\mathcal{D}|) + \log(\eta\,|\mathcal{D}|) = 0. \end{aligned}$$

3.2.1 Finding the Network Topology

If no prior knowledge about the body structure of the robot exists, we initialize a fully connected kinematic model containing a total of $\sum_{k=0}^{m} \binom{n}{2}\binom{m}{k}$ local models (linking m action signals to n relative transformations). Given a set of observations, the robot first eliminates those local models that are highly inconsistent with the data by evaluating $valid_{\mathcal{M}}(\mathcal{D})$ as described above. The remaining set of valid models is typically still large. Certain ambiguities will, for instance, remain even after infinitely many training samples. If, for example, $p(\mathbf{z}_{12} \mid q_1, \mathcal{M}_1)$ has been determined to be a valid local model, then $p(\mathbf{z}_{12} \mid q_1, q_2, \mathcal{M}_2)$ will also be. Although these alternative models might not be distinguishable regarding their data likelihood $p(\mathcal{D} \mid \mathcal{M})$, they differ significantly in their complexities k and therefore in their model quality $q(\mathcal{M})$.

To find the best topology on a global level, we aim to select the minimal subset $\hat{\mathbb{M}} \subset \mathbb{M}_{\text{valid}}$ from the superset of all valid local models $\mathbb{M}_{\text{valid}} = \{\mathcal{M}_1, \ldots \mathcal{M}_N\}$ that covers all body parts and simultaneously maximizes the overall model fit, i.e.,

$$\hat{\mathbb{M}} := \arg\max_{\mathbb{M}} \sum_{\mathcal{M} \in \mathbb{M}} q(\mathcal{M}). \tag{12}$$

This subset can be found efficiently by computing the minimal spanning tree of $\mathbb{M}_{\text{valid}}$ taking the negative model quality measure of the individual local models as cost function. For our purposes, the spanning tree needs to cover all body parts but not necessarily all action variables, since some of them might not have an influence on the robot.

To connect all n body poses in the Bayesian network, exactly $|\hat{\mathbb{M}}| = (n-1)$ local models need to be selected. This yields $\binom{|\mathbb{M}_{\text{valid}}|}{|\hat{\mathbb{M}}|}$ possible network structures to be considered. In the typical case, where the robot is composed of $n-1$ arbitrarily connected 1-DOF joints, this number reduces to the order of $O(n^3)$. Regarding the scalability to higher degrees of freedom and longer kinematic chains, the growth of the search space is of less practical importance than other factors such as the *observability* of local transformations (from a given camera view point).

We illustrate our approach with an example. Figure 6 shows a simulated robot consisting of two body parts $\mathbf{x}_1$ and $\mathbf{x}_2$ linked by a 2-DOF spherical joint with two action signals $\mathbf{q} = (q_1\ q_2)^T$. To learn its kinematic model, the robot repeatedly samples random actions $\mathbf{q}$ and sends these to its joint. After the motion comes to rest, the robot observes the resulting pose of its body parts and adds the action-pose pair to the sequence of training data. Given these pose observations, it learns four local models relating its two body parts, for all possible dependencies on the two action signals: the first model is indepedent of any action signal, the second model depends on q_1, the third model on q_2, and the fourth model on

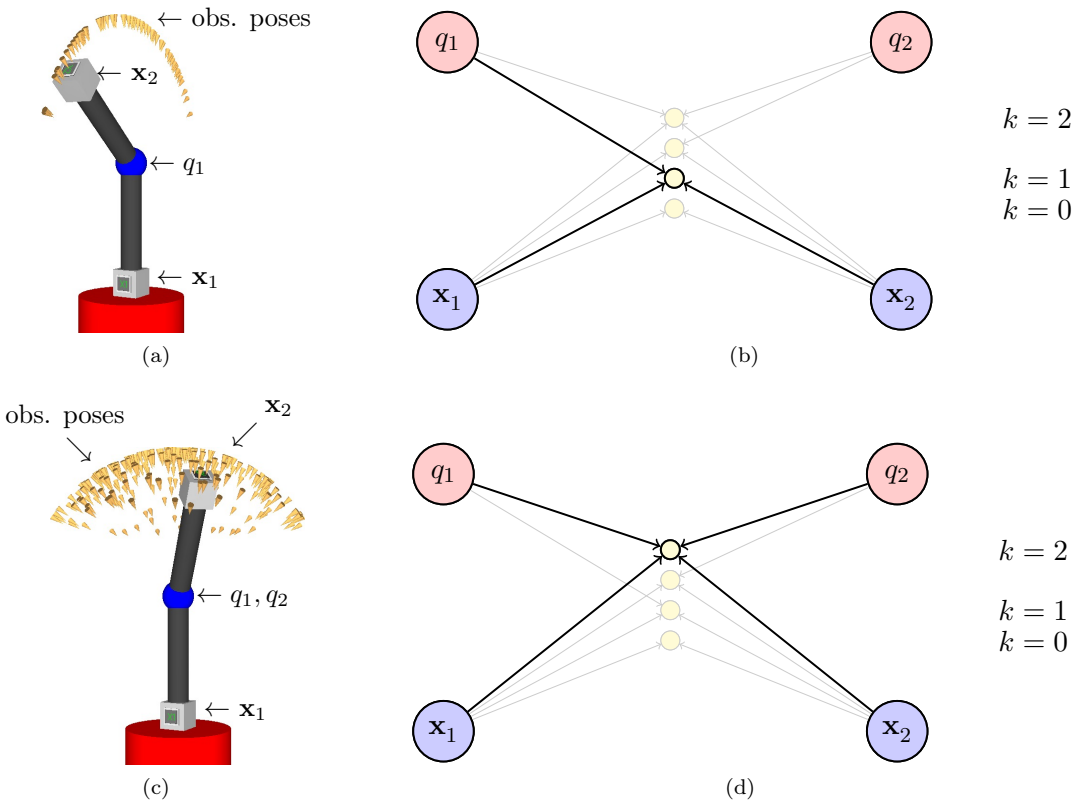


Fig. 6: Example of a 2-DOF robot composed of two body parts and a single spherical joint. (a)+(b) Result after actuating only the first DOF. (c)+(d) Result after actuating both DOF.

both action signals. Initially, we let the robot only actuate the first DOF q_1 and keep $q_2 = 0$ fixed. Correspondingly, the robot moves its end effector on a circular arc, as visualized by the yellow cones above the robot in Figure 6a. From this data, the robot trains all four local models. After learning, both models $\mathcal{M}_2$ and $\mathcal{M}_4$ are evaluated to be valid, i.e., have $\log p(\mathcal{D} \mid \mathcal{M}) > \eta|\mathcal{D}|$. With respect to our quality measure, however, $\mathcal{M}_4$ has a much higher complexity penalty as $k_2 = 1$ and $k_4 = 2$, and correspondingly, $\mathcal{M}_2$ is selected. The resulting kinematic structure is visualized by the bold arrows in Figure 6b. This situation looks different when the robot actuates both DOFs simultaneously. The resulting area covered by the end effector then corresponds to a hemisphere, as visualized in Figure 6c. Again, the robot trains all possible local models, but now finds that only $\mathcal{M}_4$ is valid (see Figure 6d). As $\mathcal{M}_2$ does not depend on the second DOF, its data likelihood is far below the acceptance threshold η and thus gets rejected. These two examples demonstrate that our quality measure favors simple models over more complex ones, but also selects more complex models if necessary.

Note that for implementing this structure search efficiently, typically not every of the $\sum_{k=0}^{m} \binom{n}{2}\binom{m}{k}$ possible local models needs to be evaluated. By the choice of the quality measure in Eq. (11), a valid model with a lower complexity will always have a higher quality than any other valid model with a larger complexity. This follows from the threshold on valid models which serves as a lower bound on the model quality: all models with data likelihoods below this threshold are invalid and thus discarded. As a consequence, an efficient algorithm can be devised to minimize the number of models to be evaluated. It is sufficient to evaluate only the first k complexity layers of local models until a minimal spanning tree is found for the first time. This spanning tree then corresponds to the global maximum of the overall model quality. The resulting algorithm is given in Algorithm 1. Important for the efficiency is that only the minimal set of local models actually gets trained and evaluated (line 2–3) and that the algorithm stops training more models after the first spanning tree has been found (line 4–6).

We illustrate the effect of this property in Figure 7. In this experiment, we consider a manipulator consisting of five body parts and four action signals. The yellow nodes correspond to all theoretically possible local models. The local models depicted in this figure are sorted corresponding to their com-

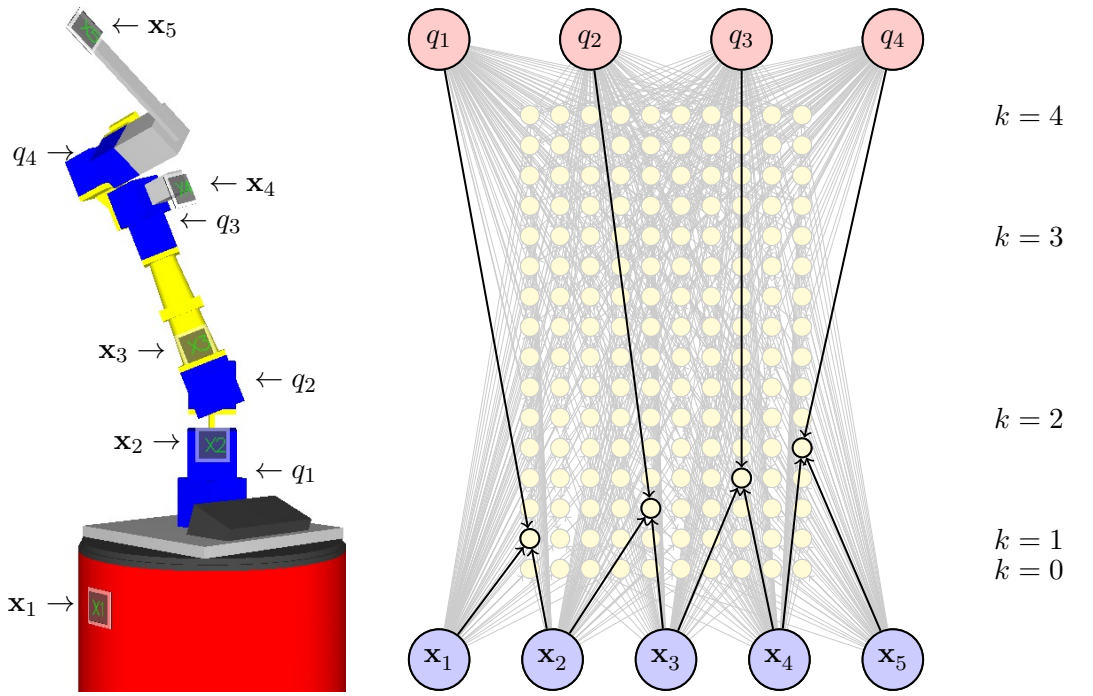


Fig. 7: (a) Example of a 4-DOF serial chain manipulator consisting of five body parts. (b) Recovered kinematic model.

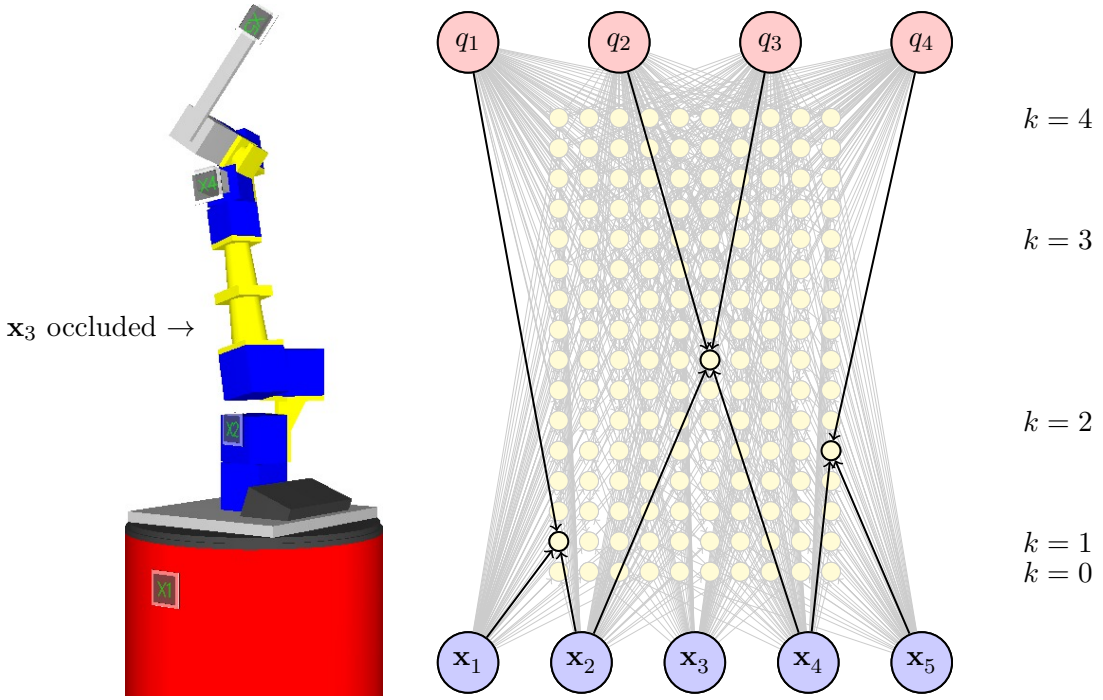


Fig. 8: Same robot as in Figure 7, but $\mathbf{x}_3$ was occluded and thus never observed. As a result, a joint model from $\mathbf{x}_2$ to $\mathbf{x}_4$ depending both on q_2 and q_3 is selected.

Input: training data $\mathcal{D}$
Output: kinematic structure $\hat{\mathbb{M}}$
for $k \in \{0, 1, \ldots, m\}$ **do**
  Let $\mathbb{M}_k := \{\mathcal{M} \mid \log p(\mathcal{D} \mid \mathcal{M}) > \eta|\mathcal{D}| \wedge |\mathcal{Q}| = k\}$ be the set of all valid models of complexity k;
  Let $\mathbb{M}_{1:k} := \bigcup_{i=1}^{k} \mathbb{M}_i$ be the set of all valid models found so far;
  if *a spanning tree of* $\mathbf{x}_1, \ldots, \mathbf{x}_n$ *exists in* $\mathbb{M}_{1:k}$ **then**
    Compute the minimum spanning tree $\hat{\mathbb{M}}$ from $\mathbb{M}_{1:k}$, for example, using Prim's or Kruskal's algorithm;
    Return $\hat{\mathbb{M}}$ as the optimal kinematic structure;
  end
end

Algorithm 1: Estimation of the kinematic structure

plexity, i.e., the bottommost row corresponds to local models representing rigid transforms ($k = 0$), the four next rows correspond to local models that depend only on a single action signal ($k = 1$), the next six rows to models that depend on two action signals simultaneously ($k = 2$), and so on. After the robot has evaluated the first two complexity layers ($k = 0$ and $k = 1$), it detects that the set of valid models contains a spanning tree, and thus the evaluation of all remaining local models with $k \geq 2$ can be skipped. The best kinematic model corresponds to the minimum spanning tree between all body parts and the local models and is visualized by the bold edges in the figure. This experiment illustrates that the proposed quality measure contributes to the efficiency of our approach, as only the first two layers of local models need to be evaluated to find the optimal kinematic model.

In a second experiment, we occluded the visual marker corresponding to the third body part of the same robot. Figure 8 shows the resulting Bayesian network. As $\mathbf{x}_3$ was never observed, no local model relating the other body parts to $\mathbf{x}_3$ could be trained. Therefore, after evaluating the local models with complexities $k = 0$ and $k = 1$, no spanning tree exists, as no valid connection between $\mathbf{x}_2$ and $\mathbf{x}_4$ can be established. Only after evaluating additionally all local models that simultaneously depend on two action signals, the robot finds a local model between $\mathbf{x}_2$ and $\mathbf{x}_4$ depending both on $\mathbf{q}_2$ and $\mathbf{q}_3$. This experiment demonstrates that our approach also works when only parts of the system are observable. However, learning local models with high-dimensional inputs is a more complex learning problem and usually requires more training samples before the same prediction accuracy is achieved.

3.3 Prediction and Control

Having discussed the learning of local models and the selection of the network structure, we now show how the resulting model can be used to predict the pose of the robot for a given action (forward kinematics) and how to infer a suitable action that moves the manipulator to a given pose (inverse kinematics).

The *kinematic forward model* can be constructed directly from the local models contained in $\mathbb{M}$, since these form a tree over all body part variables $\mathbf{x}_i$. We can write

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid q_1, \ldots, q_m) = \prod_i p(\mathbf{x}_i \mid \text{parents}(\mathbf{x}_i)) \tag{13}$$

$$= p(\mathbf{x}_{\text{root}}) \prod_{\mathcal{M}_{ij} \in \mathbb{M}} p(\Delta_{ij} \mid \mathcal{Q}_{ij}, \mathcal{M}_{ij}) \tag{14}$$

$$= p(\mathbf{x}_{\text{root}}) \prod_{\mathcal{M}_{ij} \in \mathbb{M}} p(\mathbf{x}_i^{-1}\mathbf{x}_j \mid \mathcal{Q}_{ij}, \mathcal{M}_{ij}) \ , \tag{15}$$

where $\mathbf{x}_{\text{root}}$ is the position of the robot trunk, which serves as the reference frame for all other body parts. We use $\mathcal{M}_{ij}$ to denote the local model of $\mathbb{M}$ which describes the transformation between $\mathbf{x}_i$ and $\mathbf{x}_j$. From $p(\mathbf{x}_1, \ldots, \mathbf{x}_n | q_1, \ldots, q_m)$ in the factorized form, we can now approximate the maximum likelihood estimate of the resulting body posture given an action $\mathbf{q}$ by concatenating the geometric transformations of the individual geometric transformations. We define the kinematic function by finding the maximum of the probability distribution

$$f(\mathbf{q}) := \max_{\mathbf{x}_{\text{ee}}} p(\mathbf{x}_n \mid q_1, \ldots, q_m, \mathbf{x}_{\text{root}}), \tag{16}$$

where $\mathbf{x}_{\text{ee}}$ denotes the body part corresponding to the end effector (i.e., the body part to be controlled). As all local models evaluated for a particular action $\mathbf{q}$ provide a Gaussian distribution in pose space, the marginal over the pose of the end effector can efficiently be computed as the concatenation of the marginals of the individual local models. In particular, we are interested in the maximum likelihood estimate for the end effector which we can compute efficiently by concatenation, i.e.,

$$f(\mathbf{q}) := f_{\mathcal{M}_{12}}(\mathcal{Q}_{12}) f_{\mathcal{M}_{23}}(\mathcal{Q}_{23}) \cdots f_{\mathcal{M}_{(n-1)n}}(\mathcal{Q}_{(n-1)n}). \tag{17}$$

Here, $f_{\mathcal{M}_{ij}}(\mathcal{Q}_{ij})$ refers to transformation predicted by the local model $\mathcal{M}_{ij}$ and evaluated for relevant part of the action signal $\mathcal{Q}_{ij}$. Note that also the covariances of the pose estimate can be computed efficiently by approximating the result of the multiplication of two Gaussians with a Gaussian. As each regression function $f_{\mathcal{M}_{ij}}$ corresponds to a Gaussian process, also the expected variance is known and can be propagated efficiently, similar to Eq. (17), through the Bayesian network. We may refer the interested reader to Ware and Lad [2003] on this topic. In practice, however, we found that estimating the variance directly from the training data is more reliable, as it provides us with a global estimate of the uncertainty instead of a summation over local uncertainties.

The ordering of multiplications in Eq. (17) depends on the kinematic structure defined by $\hat{\mathbb{M}}$. This ordering can efficiently be computed for example using Dijkstra's algorithm to find the (shortest) path between two nodes in the spanning tree.

In principle, the *inverse kinematic model* can be derived by applying Bayes' rule,

$$p(q_1, \ldots, q_m \mid \mathbf{x}_1, \ldots, \mathbf{x}_n) = \frac{p(q_1, \ldots, q_m)}{p(\mathbf{x}_1, \ldots, \mathbf{x}_n)}\ p(\mathbf{x}_1, \ldots, \mathbf{x}_n \mid q_1, \ldots, q_m), \tag{18}$$

it is in practice difficult to determine the maximum likelihood (ML) solution for the action $q_1, \ldots, q_m$. This is due to the fact that the target posture is typically not fully specified for all body parts but rather for the root part and the end effector. Thus, the Bayesian network is only constrained at both "ends", which results in a high-dimensional optimization problem.

For this reason, we resort to *differential kinematics* which uses the Jacobian to compute a configuration that moves the end effector iteratively towards the desired target pose Sciavicco and Siciliano, 2000. Since all individual functions $f_{\mathcal{M}_i}$ are continuous, the maximum likelihood estimate f from Eq. (17) of the forward kinematic model is continuous, too, and so the Jacobian of the forward model can be computed as

$$J_f(\mathbf{q}) = \left[\frac{\partial f(\mathbf{q})}{\partial q_1}, \ldots, \frac{\partial f(\mathbf{q})}{\partial q_m} \right]^T. \tag{19}$$

Given the Jacobian $J_f(\mathbf{q})$, it is straight-forward to implement a gradient descent-based algorithm that continuously minimizes the distance function and, thus, controls the manipulator towards the target pose. While such a "greedy" controller may get trapped in local minima of the distance function and might fail to plan around obstacles, it is often used in practice for manipulator control and forms the basis of many higher-level path-planning algorithms such as probabilistic road-maps or rapidly-exploring random trees LaValle [2006].

4 Failure Awareness and Life-Long Adaptation

So far, we have assumed that the kinematics of the robot remain unchanged during its life-time. It is clear, however, that in many real-world applications, the kinematics of a robot will change over the course of its life-time. This can, for example, be caused by material fatigue, wear and tear, or inaccurate repairs. This requires that the robot revises parts of its internal model over time and can discriminate between earlier and more recent observations to reason about such changes. We would like the robot to

detect changes of its kinematics by testing the validity of its local models continuously. It might even be useful for the robot to maintain multiple body schema at different time scales. Consider, for example, a robot that uses an accurate pre-programmed model over a long period of time and that has the ability to learn additional models in response to kinematic changes. Such a situation is depicted in Figure 9. In this experiment, we changed the tool in the end effector without notifying the system. The task of the robot is to detect this change and to learn a replacement for the mismatching local model.

To deal with model changes over time, we add a time index T to the local models $\mathcal{M}^T$ to indicate this dependency. Consequently, the size of the learning problem grows exponentially in time yielding the immense upper bound of $\sum_{k=0}^{m} \binom{n}{2}\binom{m}{k}2^{|T|}$ local models to be considered. As it is intractable to evaluate all of these local models even for small periods of time, we make three additional assumptions such that an efficient algorithm for online applications can be implemented:

1. Changes to the kinematic structure and/or kinematic properties are relatively rare events.
2. Changes happen incrementally.
3. Whatever local models were useful in the past, it is likely that similar – or even the same – local models will be useful in the future.

Due to the first assumption, we do not have to re-learn the local models continuously and re-optimize the network, but rather it is sufficient to monitor the data likelihood of the models until one of them is not evaluated as being *valid* any more. In this case, the second assumption states that the network cannot change completely at a given time step, but that we can recover the new structure by exchanging non-valid local models by re-learned ones individually. Furthermore, according to our third assumption, it is reasonable to begin the search for new models with those that are similar to previously useful models, i.e., to keep a history of successful local models and to start searching within this history before learning new models from scratch.

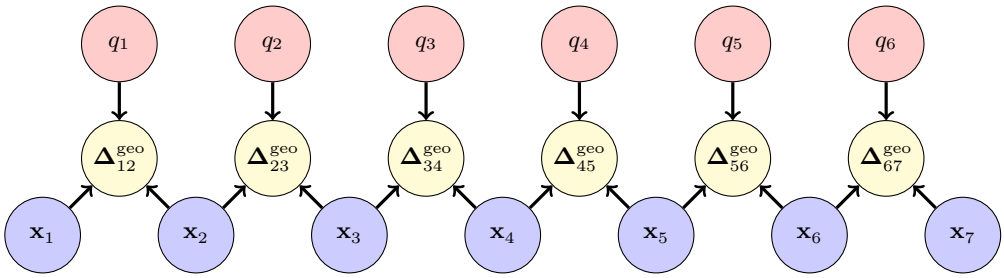
We incorporate these assumptions into an integrated system that is able to learn a body schema from scratch and to exchange local models at a later stage whenever a misfit is detected. For rating and ordering alternative local models, we consider the *structural proximity* $d_{\mathrm{DBN}}(\mathcal{M}_1, \mathcal{M}_2)$ of two local models which we define as the ratio of shared nodes in the Bayesian network. This way, models that depend on a similar set of variables are given preference in the search. We now present an experimental evaluation of the integrated system in simulation and on two real robotic manipulators.

5 Experiments

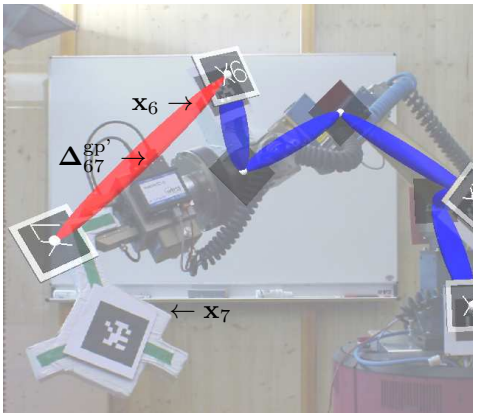
We tested our approach in a series of experiments on a real robot as well as in simulation. The goal of our experiments was to verify that

1. the robot is able to learn its kinematic structure and individual transformation functions,
2. subsequent changes to the robot's body are detected reliably (blocked joints/deformations),
3. the body schema is updated automatically without human intervention, and
4. the resulting model allows for accurate prediction and control.

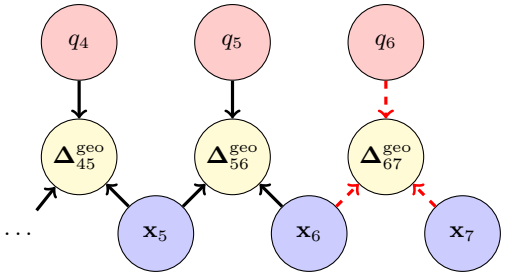
The two real robots used to carry out the experiments were equipped with a 2-DOF and with a 7-DOF manipulator, respectively, composed of Schunk PowerCube modules (see Figure 2). We compare the learned kinematic model with a carefully hand-tuned model that uses the joint encoder measurements for predicting the current pose. Note that our approach uses in contrast only the actions and not proprioception for learning the model and predicting the pose. Visual perception was implemented using a Sony DFW-SX900 FireWire camera at a resolution of 1280x960 pixels. Seven black-and-white markers were attached to the joints of the robot and the ARToolkit vision module [Fiala, 2005] was used to continuously estimate their 3D poses. The standard deviation of the camera noise was measured to $\sigma_{\mathrm{markers}} = 0.044\,\mathrm{m}$ in 3D space, which is acceptable considering that the camera was approximately located two meters away from the robot. The prediction errors and error bars reported in the following were evaluated using independent test sets $\mathcal{D}_{\mathrm{testing}}$ with 15 data samples.



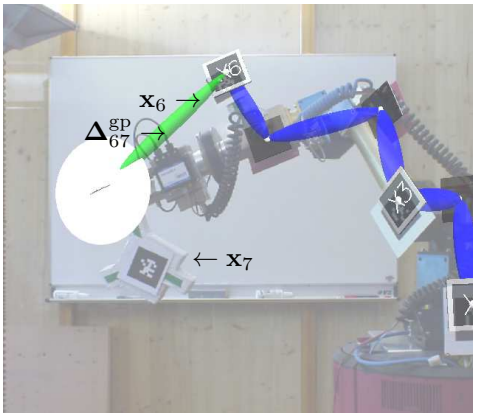
(a)



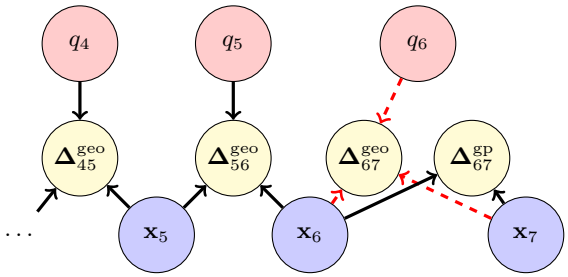
(b)



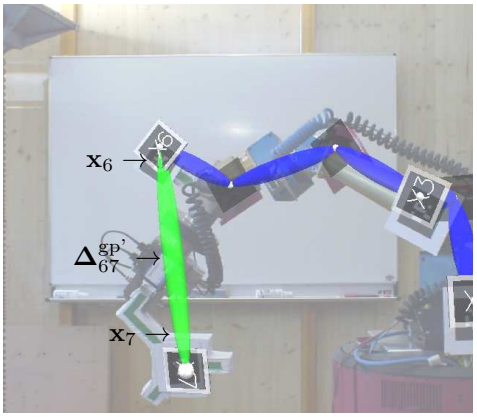
(c)



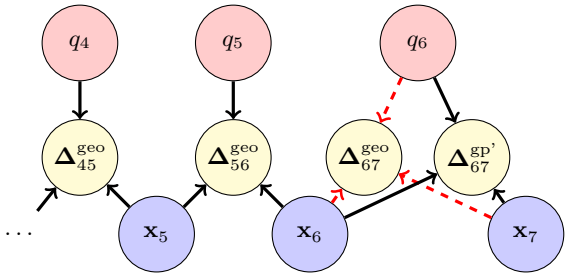
(d)



(e)



(f)



(g)

Fig. 9: Adaptation of the body schema during tool-use. (a) Initial body schema. (b) After a different tool is placed in the gripper, the model does not fit the observations anymore. (c) The mismatching model $\mathbf{\Delta}_{67}$ is revoked. (d)+(e) The first newly sampled model ($\mathbf{\Delta}^{gp}_{67}$) has a high uncertainty because of the missing dependency on the action signal q_6. (f)+(g) The second sampled model ($\mathbf{\Delta}^{gp'}_{67}$) is a suitable replacement.

5.1 Evaluation of Model Accuracy

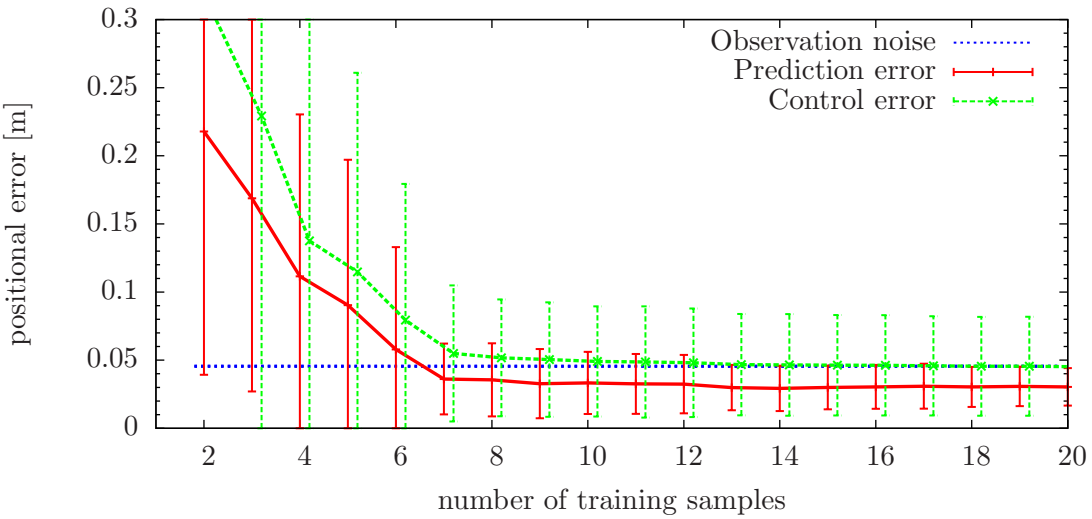
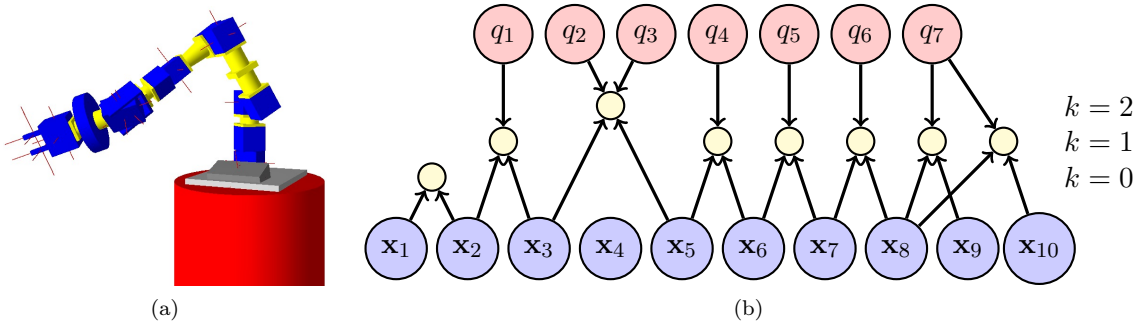


Fig. 10: Prediction and control errors of our model learning approach evaluated on a real 2-DOF manipulation robot.

To quantitatively evaluate the accuracy of the kinematic models learned from scratch as well as the convergence behavior of our learning approach, we generated random action sequences and analyzed the intermediate models using the 2-DOF robot of which the kinematic model is perfectly known. Figure 10 gives the absolute errors of prediction and control after certain numbers of observations have been processed. For a reference, we also give the average observation noise, i.e., the absolute localization errors of the visual markers. As can be seen from the diagram, the body schema converges robustly within the first 10 observations. After about 15 training samples, the accuracy of the predicted body part positions becomes even higher than 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 system is able to "blindly" estimate its pose more accurately than immediate perception. The figure also gives the accuracy when the robot is using the learned model to control its position. Here, we used an additional marker to define the target location of the end effector. We learned the full body schema from scratch as in the previous experiment and used the gradient-based control algorithm to bring the end effector to the desired target location. The average positioning error is in the order of the perception noise (approximately 0.050 m, see Figure 10), which is slightly higher than the prediction error alone.



(a) (b)

Fig. 11: Experiment with a simulated 7-DOF-manipulator consisting of 10 body parts. Body part $\mathbf{x}_4$ was occluded and, thus, never observed. (a) Picture of the simulated robot. (b) After 10 training samples, the Bayesian network has converged to the correct kinematic structure.

The second experiment was carried out on a 2-DOF robot of similar size in simulation. Therein, we analyzed the convergence behavior of the local models with respect to the training size in the absence of observation noise. We evaluated the accuracy of the learned models on independently drawn test sets. Here, we found that the accuracy was on average below 0.002 m and 1° after 20 training samples, and below 0.001 m and 0.2° after 100 training samples. This shows that the underlying Gaussian process regression models can approximate the kinematic function arbitrarily well, given that enough training data is available.

Further, we evaluated our algorithm on a simulated 7-DOF manipulator consisting of 10 body parts, to verify that our approach also scales to larger manipulators. The total length of the simulated manipulator was 1.300 m. The manipulator has been assembled as follows (see Figure 11):

- Body parts $\mathbf{x}_1$ and $\mathbf{x}_2$ were firmly connected to each other.
- Two fingers $\mathbf{x}_9$ and $\mathbf{x}_{10}$ were mounted on the 1-DOF gripper whose configuration is given by q_7.
- The remaining body constituted a chain of visible body parts $\mathbf{x}_2, \ldots, \mathbf{x}_8$ and revolute joints $q_1, \ldots, q_6$.

The structure of the learned forward model converges after around 10 samples, similar to previous experiments. The average prediction error after around 100 samples was below 0.001 m.

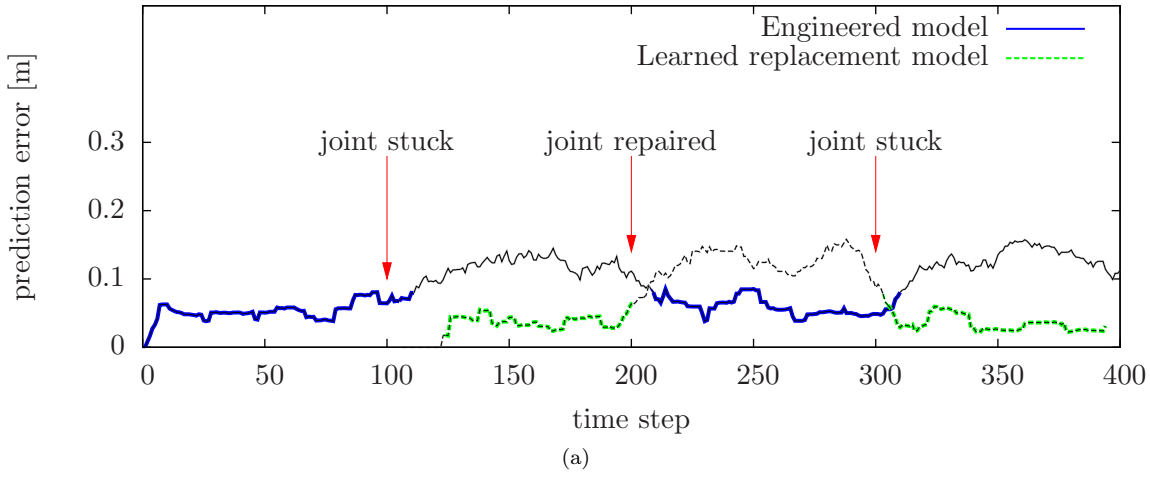
With these experiments, we demonstrated that our approach is able to recover the kinematic model of several real and simulated manipulators. Furthermore, we showed that the learned models are more accurate than the observation noise in the real robot experiment and asymptotically converge towards zero error in the noise-free case. Finally, with the experiment on the simulated 10 part manipulator and 7 DOFs, we demonstrated that our approach applies also to more complex structures.

5.2 Recovery from a Blocked Joint

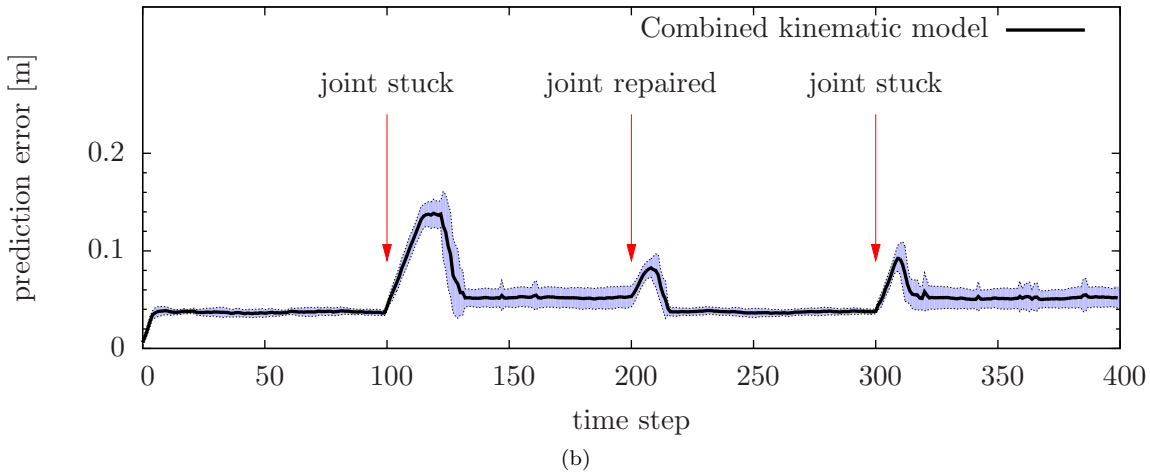
In a second experiment we used the 7-DOF robot depicted in Figure 2b to evaluated how well the proposed system can detect a stuck joint and repair its model accordingly. To this end, we initialized the body schema with an accurate, manually calibrated model. Upon detection of a model mismatch, new local models were trained from a set $\mathcal{D}_{\text{training}}$ of 30 consecutive training samples recorded after the model was instantiated. In order for a local model to be valid, its translational and orientational error on the test set was required to be within $3\sigma_{\mathbf{z},\text{pos}} = 0.150\,\text{m}$ and $3\sigma_{\mathbf{z},\text{orient}} = 45°$, with $\sigma_{\mathbf{z},\text{pos}}$ and $\sigma_{\mathbf{z},\text{orient}}$ the standard deviations of the positional and orientational observation noise, respectively. New local models were only sampled when no valid spanning tree could be constructed for 15 consecutive time steps. This corresponds to the time it takes to replace the data samples in the test set – depending on the visibility of the individual markers.

We generated a large sequence of random actions $\langle \mathbf{q}_1, \ldots, \mathbf{q}_t \rangle$. Before accepting a pose, we checked externally that these actions would not cause any (self-)collisions and that the visual markers of interest would potentially be visible on the monocular camera image. This sequence was sent to the robot and after the motion of the manipulator stopped, the observed marker poses $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ were recorded. We allowed for arbitrary motion patterns (only constrained by the geometry of the manipulator) and thus do not require full visibility of the markers. In the rare case of an anticipated or actual (self-)collision during execution, the robot stopped and the sample was rejected. Analysis of the recorded data revealed that, on average, the individual markers were visible only in 86.8 % of the images. In a second run, we blocked the end effector joint q_4 so that it could not move and again recorded a log-file. An automated test procedure was then used to evaluate the performance and robustness of our approach. For each of the 20 recorded runs, a data sequence was sampled from the log-files, consisting of 4 blocks with $N = 100$ data samples each. The first and the third block were sampled from the initial body shape, while the second and the fourth block were sampled from the log-file where the joint got blocked.

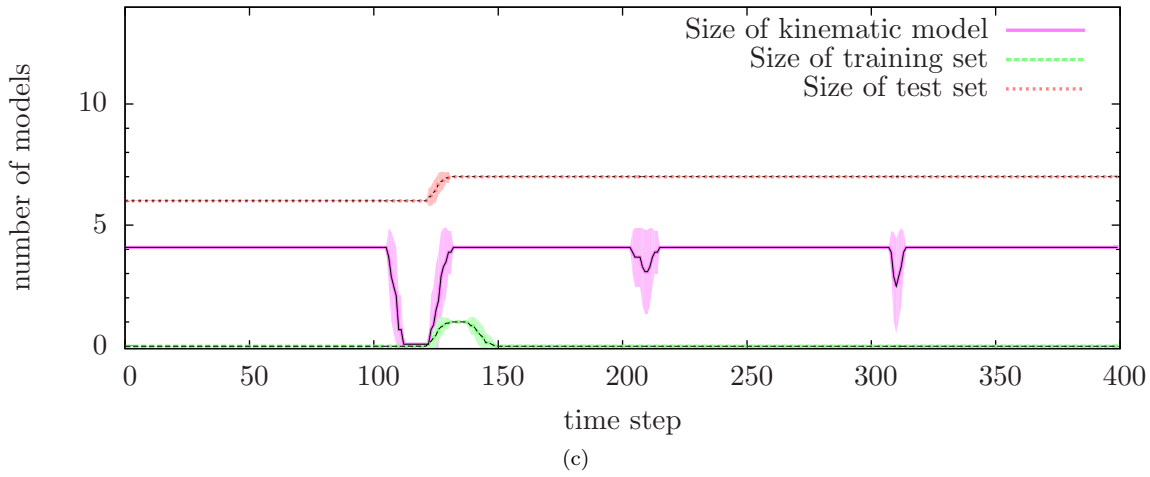
Figure 12a shows the absolute errors of the local models predicting the end effector pose. As expected, the prediction error of the engineered local model increases significantly after the end effector joint gets blocked at $t = 100$. After a few samples, the robot detects a mismatch in its internal model and starts to learn a new dynamic model (around $t = 130$), which quickly reaches the same accuracy as the original, engineered local model. At $t = 200$, the joint gets repaired (unblocked). Now the estimated error of the replacement model quickly increases while the estimated error of the engineered local model decreases



(a)



(b)



(c)

Fig. 12: Experimental evaluation of model recovery after a joint is blocked. (a) Prediction errors of the engineered and learned replacement model of a single run. (b) Prediction error of the combined model averaged over 20 runs. (c) Number of models in the current Bayesian network, the current training set, and the current test set. On average, our approach only needs to sample a single model before the kinematic model is restored.

rapidly towards its initial accuracy. Later, at $t = 300$, the joint gets blocked again in the same position, the accuracy of the previously learned replacement model increases significantly, and thus the robot can re-use this local model instead of having to learn a new one.

We averaged the precision of the combined model – i.e., the engineered one fused with the one learned after having detected the failure – over 20 runs of the experiment. The results are given in Figure 12b. The hand-tuned initial geometrical model evaluates to an averaged error at the end effector of approximately 0.037 m. After the joint gets blocked at $t = 100$, the error in prediction increases rapidly. After $t = 115$, a single new local model gets sampled, which already is enough to bring down the overall error of the combined kinematic model to approximately 0.0051 m. Training of the new local model is completed at around $t = 135$.

Later, at $t = 200$, when the joint gets un-blocked, the error estimate of the combined kinematic model increases slightly, but returns much faster to its typical accuracy: switching back to an already known local model requires less data samples than learning a new model (see Table 1). At $t = 300$, the same quick adaption can be observed when the joint gets blocked again.

Table 1: Evaluation of the number of pose observations required until the robot can re-establish a valid kinematic model after being exposed to different types of failures. The numbers give the mean and standard deviations in 20 independent runs.

Visibility rate	Failure type	Time steps until recovery		
		first occurrence	restore/repair	second occurrence
91.9 %	Joint stuck	16.50 ± 1.20	0.45 ± 0.86	0.65 ± 1.15
79.0 %	Tool exchange	20.20 ± 1.96	11.10 ± 0.83	12.10 ± 1.64

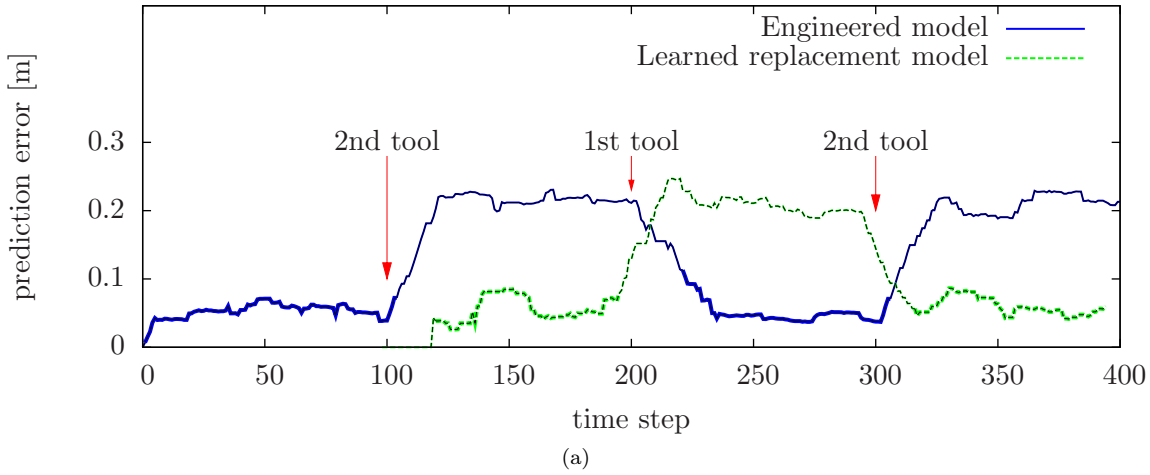
5.3 Tool Use

In a third experiment, we changed the end effector link length and orientation and applied the same evaluation procedure as in the previous subsection. This was accomplished by placing a tool with an attached marker in the gripper and changing its configuration during the experiment (see Figure 9). After a different tool is placed in the gripper, the body schema does not fit the observations anymore. In particular, the robot identifies $\mathbf{\Delta}_{67}$ as the mismatching component and seeks for a replacement. The first newly sampled model ($\mathbf{\Delta}_{67}^{gp}$) has a high uncertainty because of the missing dependency on the action signal q_6. Accordingly, the robot samples a second model $\mathbf{\Delta}_{67}^{gp'}$ which it evaluates as a suitable replacement. As a result, the adapted body schema is again valid and the robot can position its tool accurately.

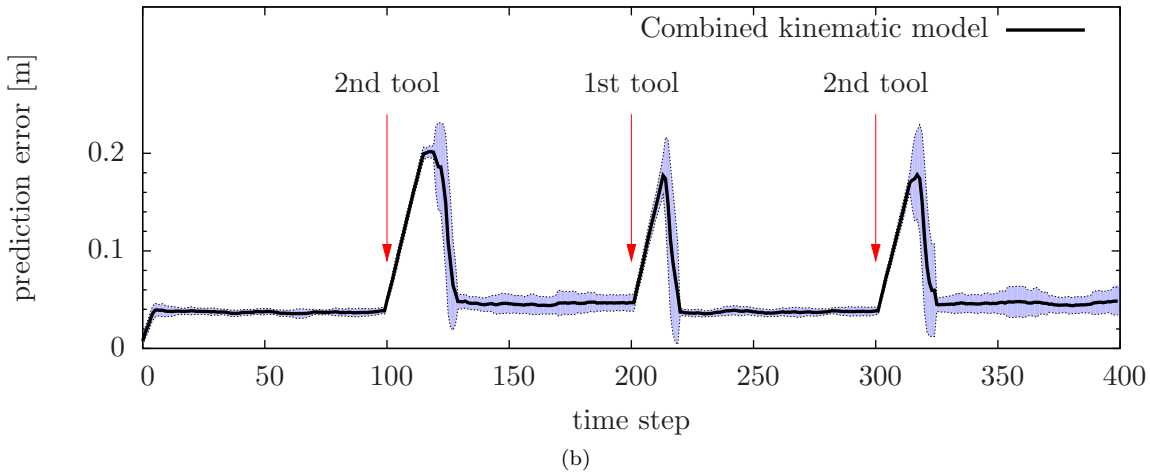
The quantitative results for a single run and the average over 20 runs of this experiment are given in Figure 13. After the tool gets displaced at $t = 100$, two local models have to be sampled on average to repair the kinematic model. The prediction accuracy of the whole system closely resembles the levels that were obtained in the case of the blocked joint. On average, we measured an accuracy of 0.047 m after recovery. In Table 1, we summarize the averaged recovery times for this and the previous experiment. As can be seen from the results, the system recovers from a blocked joint quicker than from a tool exchange, and recalling a previously successful model is significantly faster than learning a new model from scratch.

5.4 Controlling a Deformed Robot

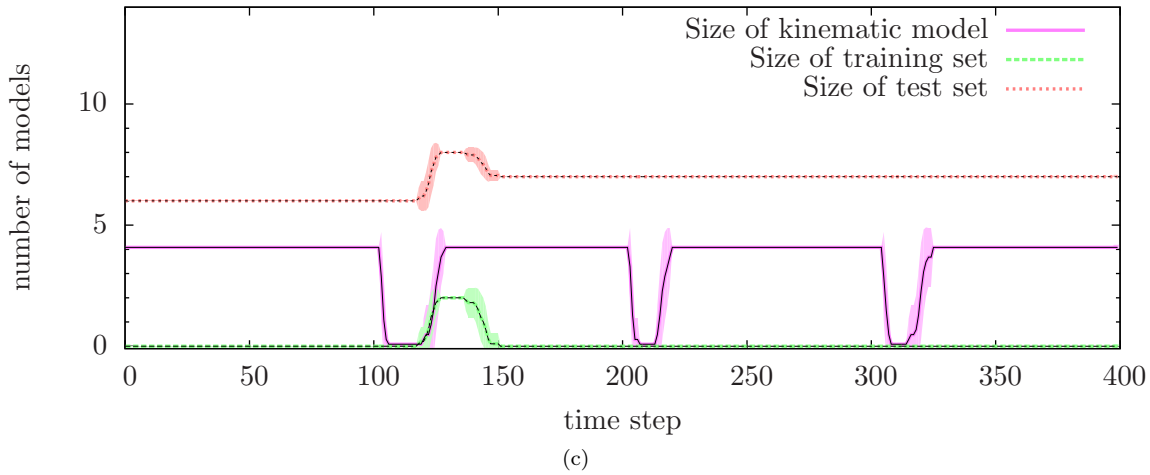
Finally, we performed a series of experiments to verify that dynamically maintained body schemata can be used for accurate positioning and control. The experiments were performed on a simulated 4-DOF manipulator. We defined a trajectory consisting of 30 way points (in 3D space) that the manipulator



(a)



(b)



(c)

Fig. 13: In this experiment, the tool in the end effector of the robot was repeatedly exchanged. (a) Prediction error of the engineered and learned replacement model of a single run. (b) Prediction error of the combined model averaged over 20 runs. (c) Evolution of models being trained and tested while the kinematic model gets updated. In this case, the robot samples on average two local models before the kinematic model is restored.

was requested to approach using the differential kinematics using its current body schema. When the initial geometric model was used to follow the trajectory by using the undamaged manipulator, the robot achieved a positioning accuracy of 0.007 m. After we had deformed the middle link by 45°, the manipulator with a static body schema was significantly off course, leading to an average positioning accuracy of 0.189 m. With dynamic adaptation enabled, the precision settled at 0.015 m. These results are also summarized in Table 2 including the standard deviations of the errors computed over 20 independent runs. The results show that dynamic model adaptation enables a robot to maintain a high positioning accuracy even after substantial changes to its kinematics.

Table 2: Evaluation of the control of a deformed robot in simulation. Experimental comparison of the control error while following a trajectory in the presence of hardware failures.

Shape	Strategy	Control error [m]
initial	static	0.007 ± 0.011
deformed	static	0.189 ± 0.028
deformed	adaptive	0.015 ± 0.002

With the experiments on blocked joints, deformed links, and tool changes, we showed that robots equipped with our approach are able to maintain a valid kinematic model even after significant damage or changes occur to the robot. Furthermore, our approach does not require to re-learn the complete model, but is able to identify inaccurate parts of the Bayesian network and to replace these efficiently using a suitable search heuristic. With our experiments on controlling a deformed robot, we demonstrated that a robot using our approach stays operational after link deformations and hardware failures and thus requires less human supervision.

6 Related Work

The central idea of our approach is to represent the kinematic model as a probabilistic Bayesian network whose vertices correspond to body parts and action signals and whose edges encode the local kinematic models. Dearden and Demiris [2005] enabled a robot to learn a Bayesian network that relates action commands to the visual motion of its gripper. In comparison to our work, the problem considered by Dearden and Demiris is much simpler as they deal only with two body parts observed in two-dimensional camera images. As a result, their model does not provide a three-dimensional kinematic model of the manipulator.

Kuipers et al. provided with the "spatial semantic hierarchy" (SSH) a set of concepts on representing and learning sensor-motor maps for robots at different abstraction levels. Their work is inspired by the concept of human cognitive maps [Kuipers and Byun, 1988, Kuipers et al., 2000, Remolina and Kuipers, 2004]. The general idea is to learn mappings that relate sensor input to motor commands and that enable a robot, for example, to follow trajectories without any prior knowledge. A different approach has been presented by Kolter and Ng [2007] who applied dimensionality reduction to find a suitable subspace in order to learn a walking gait for a four-legged robot. Another instance of approaches based on dimensionality reduction is given by the work of Grimes et al. [2006] who employed principal component analysis in conjunction with Gaussian process regression to learn walking gaits for a humanoid robot. Yoshikawa et al. [2004a] used Hebbian networks to discover the body schema from self-occlusion and self-touching sensations, and learned classifiers for body/non-body discrimination from visual data [Yoshikawa et al., 2004b]. By combining the sensor data across multiple modalities such as visual, proprioceptive and tactile sensor data, Sawa et al. [2007] enabled a robot to infer the Jacobian even for invisible hand positions.

Other approaches model the (inverse) kinematic function directly as a high-dimensional regression problem. For example, Fiala [1994] and later Natale [2004] used neural networks to learn reaching movements, Gaskett and Cheng [2003] proposed self-organizing maps to coordinate hand-eye movements, and Kumar et al. [2010] employed radial basis function networks (RBFs) to learn the local mappings

between configurations and end effector poses. Recurrent neural networks have also been used to learn the kinematics and dynamics of manipulation robots [Reinhart and Steil, 2008, Rolf et al., 2009]. As no global inverse kinematic function exists for redundant kinematic chains, D'Souza et al. [2001] estimated the inverse kinematic function locally from observed data. As the required number of training samples increases exponentially with the degrees of freedom of the robot, Lopes and Santos-Victor [2005] proposed to learn the kinematic function incrementally, first by moving only the shoulder and elbow joints and, subsequently, the hand. Other approaches aim to reduce the number of training samples required to learn an accurate kinematic model. Martinez-Cantin et al. [2010] showed how active learning can be used to reduce the number of required training samples, by actively choosing joint configurations that maximize the expected information gain. Angulo and Torras [2005] approached this problem by splitting the manipulator into two or more virtual robots. However, Angulo and Torras assumed that a suitable decomposition of the manipulator is known beforehand, and thus, did not tackle the problem of learning the kinematic structure.

The approach presented in this article is also related to the problem of parameter optimization, which can be understood as a sub-problem of body schema learning. When the kinematic model is given in a parametric form, the parameters can be optimized efficiently with respect to an error measure [Gatla et al., 2007, Pradeep et al., 2010, He et al., 2010] or the data likelihood [Roy and Thrun, 1999]. Hersch et al. [2008] showed that parameter optimization can also be used to adapt the body schema during tool-use, for example, to estimate the tool position and orientation. Martinez-Cantin et al. [2010] extended this approach to active learning, i.e., they generated observation actions that maximize the expected information gain. Such methods can also be used to identify the dynamic parameters such as the center of mass, the moments of inertia, etc. Ting et al. [2006], for example, presented a Bayesian approach for estimating these parameters on two different manipulation robots. In principle, these methods could be applied after our approach has bootstrapped the kinematic model, in order to refine or augment the model and achieve a faster convergence. Genetic algorithms can also be used for parameter optimization given a suitable parametrization of the kinematic model space. Bongard et al. [2006a,b] described a robotic system that continuously learns its own structure from actuation-sensation relationships. Their system generates new structure hypotheses using stochastic optimization, which are validated by generating actions and by analyzing the following sensory input. In a more general context, Bongard and Lipson [2007] studied structure learning in arbitrary nonlinear systems using similar mechanisms.

In contrast to all of the approaches described above, our approach learns both the structure as well as the functional mappings for the individual building blocks of the body schema. Furthermore, it does not require an explicit parametrization of the body schema, and the representation in form of a Bayesian network allows a robot to quickly revise its structure and to replace invalidated local models on-the-fly. Recently, Hoffmann et al. [2010] published a comprehensive review on body schema learning in robotics which includes a detailed discussion of our work.

7 Summary

In this chapter, we presented a novel approach to body schema learning for manipulation robots. Our central idea is to continuously learn a large set of local kinematic models using nonparametric regression. Given this set of models, we search for the arrangement that best represents the full system. Our approach recovers the kinematic structure by finding the minimum spanning tree in the set of possible models. To the best of our knowledge, this is the first time that models of such complex kinematic systems have been learned from scratch using visual self-observation. In experiments carried out with real manipulation robots and in simulation, we demonstrated that our system is able to deal with missing and noisy observations, operates in full 3D space, and allows a robot to robustly control its end effector even in the presence of hardware failures. With our approach, we contribute an innovative solution that increases the dependability and accuracy of manipulation robots that operate over extended periods of time without the supervision of an expert.

References

V.R. De Angulo and C. Torras. Using PSOMs to learn inverse kinematics through virtual decomposition of the robot. In *Proc. of the Intl. Work-Conf. on Artificial Neural Networks (IWANN)*, Barcelona, Spain, 2005.

J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proc. of the National Academy of Sciences*, 104(24):9943–9948, 2007.

J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006a.

J. Bongard, V. Zykov, and H. Lipson. Automated synthesis of body schema using multiple sensor modalities. In *Proc. of the Intl. Conf. on the Simulation and Synthesis of Living Systems*, Bloomington, IN, USA, 2006b.

S.R. Buss and J. Kim. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools*, 10(3):37–49, 2005.

J.J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Publishing Company, 1989.

A. Dearden and Y. Demiris. Learning forward models for robots. In *Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, 2005.

A. D'Souza, S. Vijayakumar, and S. Schaal. Learning inverse kinematics. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Maui, HI, USA, 2001.

J.C. Fiala. A network for learning kinematics with application to human reaching models. In *Proc. of the IEEE Intl. Conf. on Neural Networks(ICNN)*, volume 5, pages 2759–2764, 1994.

M. Fiala. ARtag, a fiducial marker system using digital techniques. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, 2005.

S. Gallagher. *How the Body Shapes the Mind*. Oxford University Press, USA, 2005.

C. Gaskett and G. Cheng. Online learning of a motor map for humanoid robot reaching. In *Proc. of the Intl. Conf. on Computational Intelligence, Robotics and Autonomous Systems (CIRAS)*, Singapore, 2003.

C.S. Gatla, R. Lumia, J. Wood, and G. Starr. An automated method to calibrate industrial robots using a virtual closed kinematic chain. *IEEE Transactions on Robotics (T-RO)*, 23(6):1105 –1116, 2007.

D. Grimes, R. Chalodhorn, and R. Rao. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In *Proc. of Robotics: Science and Systems (RSS)*, Philadelphia, PA, USA, 2006.

R. He, Y. Zhao, S. Yang, and S. Yang. Kinematic-parameter identification for serial-robot calibration based on POE formula. *IEEE Transactions on Robotics (T-RO)*, 26(3):411–423, 2010.

M. Hersch, E. Sauser, and A. Billard. Online learning of the body schema. *Intl. Journal of Humanoid Robotics*, 5(2):161–181, 2008.

M. Hoffmann, H. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R Pfeifer. Body schema in robotics: a review. *IEEE Transactions on Autonomous Mental Development*, 2(4):304–324, 2010.

J. Kolter and A. Ng. Learning omnidirectional path following using dimensionality reduction. In *Proc. of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

B. Kuipers and Y.-T. Byun. A robust, qualitative method for robot spatial learning. In *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, Saint Paul, MN, USA, 1988.

B. Kuipers, R. Browning, B. Gribble, M. Hewett, and E. Remolina. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

S. Kumar, L. Behera, and T.M. McGinnity. Kinematic control of a redundant manipulator using an inverse-forward adaptive scheme with a KSOM based hint generator. *Robotics and Autonomous Systems (RAS)*, 58(5):622–633, 2010.

S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

M. Lopes and J. Santos-Victor. Visual learning by imitation with motor representations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(3):438–449, 2005.

Angelo Maravita and Atsushi Iriki. Tools for the body (schema). *Trends in Cognitive Sciences*, 8(2): 79–86, 2004.

R. Martinez-Cantin, M. Lopes, and L. Montesano. Body schema acquisition through active learning. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Anchorage, AK, USA, 2010.

A.N. Meltzoff and M.K. Moore. Explaining facial imitation: A theoretical model. *Early Development and Parenting*, 6(3–4):179–192, 1997.

G. Metta, G. Sandini, L. Natale, L. Craighero, and L. Fadiga. Understanding mirror neurons: A bio-robotic approach. *Interaction Studies*, 7:197–232, 2006.

C. Nabeshima, Y. Kuniyoshi, and M. Lungarella. Adaptive body schema for robotic tool-use. *Advanced Robotics*, 10(20):1105–1126, 2006.

L. Natale. *Linking action to perception in a humanoid robot: A developmental approach to grasping.* PhD thesis, University of Genoa, Italy, 2004.

V. Pradeep, K. Konolige, and E. Berger. Calibrating a multi-arm multi-sensor robot: A bundle adjustment approach. In *Intl. Symp. on Experimental Robotics (ISER)*, New Delhi, India, 2010.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning.* Adaptive Computation and Machine Learning. The MIT Press, 2006.

R.F. Reinhart and J.J. Steil. Recurrent neural associative learning of forward and inverse kinematics for movement generation of the redundant PA-10 robot. In *Proc. of the ECSIS Symp. on Learning and Adaptive Behaviors for Robotic Systems (LAB-RS)*, Edinburgh, United Kingdom, 2008.

E. Remolina and B. Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152 (1):47–104, 2004.

M. Rolf, J.J. Steil, and M. Gienger. Efficient exploration and learning of whole body kinematics. In *Proc. of the IEEE Intl. Conf. on Development and Learning (ICDL)*, Shanghai, China, 2009.

N. Roy and S. Thrun. Online self-calibration for mobile robots. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Detroit, MI, USA, 1999.

F. Sawa, M. Ogino, and M. Asada. Body image constructed from motor and tactile images with visual information. *Intl. Journal of Humanoid Robotics*, 4(2):347–364, 2007.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

L. Sciavicco and B. Siciliano. *Modeling and Control of Robot Manipulators.* Advanced Textbooks in Control and Signal Processing. Springer, 2000.

M.I. Stamenov. *Body schema, body image, and mirror neurons*, chapter 2. John Benjamins Publishing, 2005.

J. Ting, M. Mistry, J. Peters, S. Schaal, and J. Nakanishi. A Bayesian approach to nonlinear parameter identification for rigid body dynamics. In *Proc. of Robotics: Science and Systems (RSS)*, Philadelphia, PA, USA, 2006.

R. Ware and F. Lad. Approximating the distribution for sums of products of normal variables. Technical Report UCDMS 2003/15, University of Canterbury, New Zealand, 2003.

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 Intl. Workshop on Epigenetic Robotics*, Genoa, Italy, 2004a.

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 Intl. Conf. on Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2004b.