

# Body Schema Learning for Robotic Manipulators from Visual Self-Perception

Jürgen Sturm<sup>\*,a</sup>, Christian Plagemann<sup>b</sup>, Wolfram Burgard<sup>a</sup>

<sup>a</sup>*University of Freiburg, Department of Computer Science, Georges-Köhler-Allee 79,  
79110 Freiburg, Germany*

*{sturm, burgard}@informatik.uni-freiburg.de*

<sup>b</sup>*Stanford University, AI Lab, 353 Serra Mall, Stanford, CA 94305-9010, USA  
plagemann@stanford.edu*

---

## Abstract

We present an approach to learning the kinematic model of a robotic manipulator arm 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. In experiments carried out both on real and simulated robotic manipulators, we verified the validity of our approach for real-world problems such as end-effector pose prediction and end-effector pose control.

*Key words:* robotics, machine learning, self-perception, sensor-motor learning, body schema

---

\*Corresponding author

URL: <http://www.informatik.uni-freiburg.de/~sturm> (Jürgen Sturm)

## 1. Introduction

Kinematic models are widely used in robotics, especially in the context of robotic manipulation (Craig, 1989; Choset et al., 2005). Such models are typically derived in an analytic way (Rosales and Gan, 2004), which relies heavily on prior knowledge about the geometry of the robot. When a model is then used on a real robotic manipulator, its geometrical parameters have to be calibrated carefully. In practice, this is done by the robot manufacturer using a highly accurate calibration chamber.

As robotic systems become more complex and versatile, especially when they are delivered in a completely reconfigurable way, there is a growing demand for techniques allowing a robot to automatically learn kinematic models with no or only minimal human intervention and without the need for costly calibration equipment. Clearly, such a capability would not only facilitate the deployment and calibration of new robotic systems but also allow for autonomous re-adaptation when the kinematic model changes, e.g., due to deformations of robot parts or due to material fatigue. Additionally, components of the robot might get exchanged or replaced by newer parts such that the system model no longer complies with the originally engineered one. Finally, to make intelligent use of tools, a robot would need to learn the kinematic properties of unknown tools to use them appropriately (Nabeshima et al., 2006).

In this article, we investigate how to equip robots with the ability to learn their kinematic models from scratch and how to adapt them over time using exploratory actions and self-perception. Figure 7 depicts the robotic manipulation system used to develop and test our approach and Fig. 7 shows

a visualization of its simulated counter-part. We propose to learn a Bayesian network for the kinematic structure of the robot including the forward and inverse models relating action commands and body pose. In contrast to prior work in this area, we jointly address (a) the combinatorial problem of assigning action signals and percepts to body parts and (b) the calibration problem for all the individual kinematic transformations. Figure 3 gives an overview of the proposed approach. We start with a fully connected network containing all perceivable body parts and the available action signals, perform random “motor babbling” and iteratively reduce the network complexity by analyzing the perceived body motion. At the same time, we learn Gaussian process regression models for all individual 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.

Our approach addresses all of the following practical problems that frequently arise in robotic manipulation tasks in a single Bayesian framework:

- **Pose prediction:** Given the current joint configuration (i.e., the joint angles), compute the pose of the end-effector.
- **Pose control:** Conversely, given a target pose (a 6D configuration of the end-effector), compute the required joint angles that lead to that position.
- **Model selection:** Given a history of observed configuration–pose pairs, infer the structure of the kinematic function. In particular, this includes discovering the (in-)dependencies between the individual body parts and the joint signals.

- **Model learning:** After the kinematic structure of the robot has been identified, the geometrical relationships between the body parts of the robot can be learned from the history of observed configuration–pose pairs. If parametrized models are used, this corresponds to parameter optimization given the observed data.
- **Model testing:** Given both a pose prediction and a pose observation, the robot estimates the quality of its internal model.
- **Failure detection and model adaptation:** When the kinematics of the robot change, e.g., when a joint gets blocked or is deformed, the robot detects such a change by model testing. Our approach can localize the mismatch within the kinematic function and trigger re-learning only for the affected parts of the kinematic function.

The article is structured as follows. We start with a brief introduction of kinematic and inverse kinematic functions in robotics in Section 2. We will then introduce our Bayesian framework for representing robotic body schemas in Section 3, discuss failure awareness and life-long adaptation in Section 4, and present experimental results obtained with real and simulated manipulator arms in Section 5, which demonstrate that our approach is able to quickly learn compact and accurate models and to robustly deal with noisy observations. We finish the article with a discussion of related work.

## 2. The Kinematic Function and the Body Schema

The kinematic model describes the relationship between the configuration of the robot, i.e., the joint angles, and the body posture, i.e., the positions

of the body parts in space. Figure 4 shows an example of a simple 2-DOF robotic manipulator. The robot consists of two rotary joints  $a_1$  and  $a_2$ , and five body parts  $\mathbf{X}_1, \dots, \mathbf{X}_5$ . The first two body parts are connected rigidly. This means that the geometric transformation  $\Delta_{1 \rightarrow 2}$  from the trunk  $\mathbf{X}_1$  to the shoulder  $\mathbf{X}_2$  is independent of the positions of the joints. The shoulder  $\mathbf{X}_2$  and the upper arm  $\mathbf{X}_3$  are connected by the shoulder joint  $a_1$ , and thus their geometric transformation  $\Delta_{2 \rightarrow 3}(a_1)$  depends on the joint angle of  $a_1$ . The same holds for the following parts, as the joint angle of the elbow joint  $a_2$  has direct influence on the geometrical transformation  $\Delta_{3 \rightarrow 4}(a_2)$  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  $\Delta_{4 \rightarrow 5}$  is a fixed transformation. The kinematic function of this manipulator is defined by the concatenation of these individual transforms, i.e.,

$$f(a_1, a_2) := \Delta_{1 \rightarrow 2} \circ \Delta_{2 \rightarrow 3}(a_1) \circ \Delta_{3 \rightarrow 4}(a_2) \circ \Delta_{4 \rightarrow 5} . \quad (1)$$

The kinematic function  $f(a_1, a_2)$  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 robotic manipulator 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 required joint angles  $a_1, a_2$  given a target position in the workspace, requiring the inverse kinematic function  $f^{-1}$ . Note that as the algebraic inversion of a kinematic function  $f$  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 pseudo-inverse inverse kinematics (IK) or damped-least

squares IK (Buss and su Kim, 2005).

Probabilistic graphical models, like Bayesian networks, are an appealing tool to represent the dependencies between random variables in an intuitive way (Pearl, 1988; Jensen, 2001). A Bayesian network is a directed, acyclic graph consisting of nodes and edges. The nodes in the network correspond to the observed or latent random variables, and the edges (or lack of edges) between nodes stand for the conditional dependencies (or independencies, respectively) between two nodes. Note how the kinematic function of Eq. 1 has been translated into the Bayesian network depicted in Fig. 7: the poses of the body parts  $\mathbf{X}_1, \dots, \mathbf{X}_5$  and the actions  $a_1, a_2$  appear as nodes in the network, and the kinematic structure is encoded as the topology of the network. Consider for example the pose of body part  $\mathbf{X}_2$ , that has only a single incoming edge originating from pose  $\mathbf{X}_1$ , meaning that the pose of  $\mathbf{X}_2$  is fully defined after  $\mathbf{X}_1$  has been observed. In contrast, the pose of  $\mathbf{X}_3$  has incoming edges both from  $\mathbf{X}_2$  and  $a_1$ , expressing its probabilistic dependency of both of its so-called parents. Bayesian networks can be used to infer the probability distributions of particular nodes in a variety of ways (Sudderth et al., 2003). In particular, it is possible with Bayesian networks to answer queries similar to those listed at the end of Section 1, for example predicting the pose of the end-effector (given  $a_1, \dots, a_m$ , infer  $\mathbf{X}_n$ ) or controlling the pose of the end-effector (given  $\mathbf{X}_n$ , infer  $a_1, \dots, a_m$ ).

### 3. A Bayesian Framework for Kinematic Chains

We define a robotic body schema as the joint probability distribution of the available action signals  $\langle a_1, \dots, a_m \rangle$ , the self-observations  $\langle \mathbf{Y}_1, \dots, \mathbf{Y}_n \rangle$ ,

and the true poses of the body parts  $\langle \mathbf{X}_1, \dots, \mathbf{X}_n \rangle$ . In our concrete scenario, in which we consider the body schema of a robotic manipulator arm in conjunction with a stationary, monocular camera, the action signals  $a_i \in \mathbb{R}$  are real-valued variables corresponding to the latest joint angle request. Whereas the  $\mathbf{X}_i \in \mathbb{R}^6$  encode the 6-dimensional poses (3D Cartesian position and 3D orientation) of the body parts w.r.t. a reference coordinate frame, the  $\mathbf{Y}_i \in \mathbb{R}^6$  are observations of the body parts—generally noisy and potentially missing. Throughout this article, we use capital, bold letters to denote the pose variables to highlight that these also uniquely define homogeneous transformation matrices, which can be concatenated and inverted.

Note that our approach does not require proprioception telling the robot how well joint  $i$  has approached the requested target angle  $a_i$ . At first sight, it seems that with proprioception one could learn the kinematic function passively from visual and proprioceptive observations only. However, one would then lack the mapping from motor commands to proprioception, 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 motor-proprioeption mapping for each joint. Therefore, we propose to learn the mapping from motor commands to body pose observations directly using active exploration. In this way, our approach closes the action-perception-loop, as visualized in Fig. 3, and it obviates the need for additional calibration of the motor encoders.

Formally, we seek to learn the probability distribution

$$p(\mathbf{X}_1, \dots, \mathbf{X}_n, \mathbf{Y}_1, \dots, \mathbf{Y}_n \mid a_1, \dots, a_m), \quad (2)$$

which in this form is intractable for all but the simplest scenarios. It is therefore typically assumed that each observation variable  $\mathbf{Y}_i$  is independent from all other variables given the true configuration  $\mathbf{X}_i$  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 body parts and action signals could be assumed, which in turn would lead to the much simpler, factorized model

$$p(\mathbf{X}_1, \dots, \mathbf{X}_n \mid a_1, \dots, a_m) = \prod_i p(\mathbf{X}_i \mid \text{parents}(\mathbf{X}_i)). \quad (3)$$

Here,  $\text{parents}(\mathbf{X}_i)$  comprises all body parts and action variables that directly influence  $\mathbf{X}_i$ . 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) transformation variables  $\Delta_{i \rightarrow j} := \mathbf{X}_i^{-1} \mathbf{X}_j$  for all pairs  $\langle \mathbf{X}_i, \mathbf{X}_j \rangle$  of body parts. We represent the 6D pose vectors  $\mathbf{X}$  as their equivalent homogeneous transformation matrices, which means that  $\Delta_{i \rightarrow j}$  reflects the (deterministic) relative transformation between  $\mathbf{X}_i$  and  $\mathbf{X}_j$ . Note that various parameterizations of such transformation matrices are possible (e.g., by means of the Euler angles, quaternions, or over-parametrized as the full 12D matrices) and that we thus assume a  $d$ -dimensional parametrization of  $\Delta_{i \rightarrow j}$ .

Denoting with  $\mathbf{Z}_{i \rightarrow j} := \mathbf{Y}_i^{-1} \mathbf{Y}_j$  the transformation relating the observations  $\mathbf{Y}_i$  and  $\mathbf{Y}_j$  that correspond to  $\mathbf{X}_i$  and  $\mathbf{X}_j$ , we define as a *local model* the subgraph of our network that defines the relationship between any two body parts  $\mathbf{X}_i$  and  $\mathbf{X}_j$  and their dependent variables, if all other body parts are ignored. Figure 7 shows a prototypical local model. Here, we denote



with  $\mathcal{A}_{i \rightarrow j}$  the set of action variables that have a direct influence on  $\Delta_{i \rightarrow j}$ . 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.

In the following, we explain (a) how to learn local models from data and (b) how to find the best spanning tree built from these local models that 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, the one action sequence would be selected that maximizes the overall gain while keeping all possible risks low (Stachniss et al., 2005). 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 Fig. 7) describes the geometric relationship between two body parts  $i$  and  $j$  given a set of action signals  $\mathcal{A}_{i \rightarrow j}$ . We propose to learn this relationship from data samples acquired while performing random actions and observing their effects. As the learning framework for solving this supervised regression problem, we apply Gaussian processes for regression (Rasmussen and Williams, 2006). On the real robotic plat-

form used in our experiments, the action  $a_i$  correspond to the target angle requested from joint  $i$ . The observations  $\mathbf{Y}_i$  of part locations  $\mathbf{X}_i$  are obtained by tracking visual markers in 6D space (including their 3D pose and 3D orientation (Fiala, 2004), see Fig. 7). Note that the  $\mathbf{Y}_i$ 's are inherently noisy and that missing observations are common—for example in the case of (self-)occlusion.

Formally, the task is to learn the local transformations  $\Delta_{i \rightarrow j}$ , each linking two body parts  $\mathbf{X}_i$  and  $\mathbf{X}_j$ . Considering Fig. 7, 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  $\mathbf{Y}_j$ , for instance assuming Gaussian white noise on the observations,  $\mathbf{Y}_i \sim \mathcal{N}(\mathbf{X}_i, \sigma_{\text{sensor}}^2 \cdot \mathbf{I}_d)$ . With  $\mathbf{I}_d$  we denote the  $d$ -dimensional identity matrix and remind that  $d$  denotes the chosen dimensionality of the parametrization of transformation matrices. In other terms, one would be following the full Bayesian pathway  $\mathbf{Y}_i \leftarrow \mathbf{X}_i \rightarrow \Delta_{i \rightarrow j} \rightarrow \mathbf{X}_j \rightarrow \mathbf{Y}_j$  to reason about  $\Delta_{i \rightarrow j}$ .

However, since the *absolute* positions  $\mathbf{X}_i$  are irrelevant for describing the *relative* transformations, we take a slightly different approach by concentrating on the transformations  $\mathbf{Z}_{i \rightarrow j}$  between *observations*  $\mathbf{Y}_i$  and  $\mathbf{Y}_j$ . We model  $\mathbf{Z}_{i \rightarrow j} \sim \mathcal{N}(\Delta_{i \rightarrow j}, \sigma_{\text{sensor-rel}}^2 \cdot \mathbf{I}_d)$ , and thus follow the shorter path  $\{\mathbf{Y}_i, \mathbf{Y}_j\} \leftarrow \mathbf{Z}_{i \rightarrow j} \leftarrow \Delta_{i \rightarrow j}$ , which does not include the  $\mathbf{X}_i$  explicitly. The problem of learning a single local model now has the form of the noisy regression problem, that is, learning the function

$$\begin{aligned} f_{\mathcal{M}} : \quad \mathbb{R}^{|\mathcal{A}_{i \rightarrow j}|} &\rightarrow \mathbb{R}^d, \\ \mathcal{A}_{i \rightarrow j} &\mapsto \Delta_{i \rightarrow j} \end{aligned} \tag{4}$$

from noisy observations  $\mathbf{Z}_{i \rightarrow j}$ .

For simplicity, we consider over-parametrized transformation matrices in the following with  $d = 12$  independent components and learn the functional mapping for each component 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 (orthonormalizing the rotation part using singular value decomposition) resolves the problem. In the future, we might consider more efficient parameterizations that come closer to the actual 6-DOF of the transformations.

For solving the regression problem as stated in Eq. 4, we place individual Gaussian process priors (Rasmussen and Williams, 2006) on the 12 variables of the transformation functions  $f_{\mathcal{M}}$  for all local models  $\mathcal{M}$  and choose the squared exponential covariance function to parametrize the process. Figures 8 and 9 shows the  $x$ ,  $y$ , and  $z$  components of two different local models learned from real data using the Gaussian process (GP) model. In the situation shown in Fig. 8, the action variable (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 Fig. 9, a local model has been learned for variables that do not have a direct physical relationship. As a result, the model shows high noise levels and it does not explain the data well. Such a local model is likely to be discarded during the search for the full body model, which is described in the following.

### 3.2. Learning a Factorized Full Body Model

We seek to find a factorized model for the whole kinematic structure (see Eq. 3) that explains the observed data well and that is not overly complex—such that it can be learned and evaluated online. To limit complexity, we first discard 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 its prediction error is below some threshold  $\theta$ , i.e.,  $\epsilon_{\text{pred}}(\mathcal{D}) < \theta$  that we will denote with the Boolean predicate  $\text{valid}_{\mathcal{M}}(\mathcal{D})$ . Our experiments revealed that a good value for  $\theta$  is  $3\sigma$ , where  $\sigma$  is the standard deviation of the sensor model. The prediction error  $\epsilon_{\text{pred}}(\mathcal{D}|\mathcal{M})$  is defined as

$$\epsilon_{\text{pred}}(\mathcal{D} | \mathcal{M}) := \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{Z}_{i \rightarrow j}, \mathcal{A}_{i \rightarrow j}) \in \mathcal{D}} \epsilon_{\text{pred}}(\mathbf{Z}_{i \rightarrow j} | \mathcal{A}_{i \rightarrow j}, \mathcal{M}) \quad (5)$$

with

$$\epsilon_{\text{pred}}(\mathbf{Z}_{i \rightarrow j} | \mathcal{A}_{i \rightarrow j}, \mathcal{M}) := \frac{1}{d} \sqrt{\sum_{z \in \mathbf{Z}_{i \rightarrow j}} (z - \mu_z^*)^2}. \quad (6)$$

Here,  $\mu_z^*$  is the mean prediction of component  $z$  of transformation  $\mathbf{Z}_{i \rightarrow j}$ . Denoting with  $C(\mathcal{M}) \in \mathbb{N}$  the dimensionality of model  $\mathcal{M}$ , that is, the number  $|\mathcal{A}_{i \rightarrow j}|$  of action signals that the model depends on, we define a *model quality measure*  $q(\mathcal{D}|\mathcal{M})$ ,

$$\log q(\mathcal{D} | \mathcal{M}) = \underbrace{\log[1/\epsilon_{\text{pred}}(\mathcal{D} | \mathcal{M})]}_{\text{accuracy}} - \underbrace{C(\mathcal{M}) \log 1/\theta}_{\text{complexity}} \quad (7)$$

which is proportional to both the model accuracy and to a penalty term for model complexity. Note that quality measures such as the Bayesian information criterion (BIC) are not directly applicable here since Gaussian process

regression is a *nonparametric* method and, thus, there is no obvious number of parameters for representing the data sets. Note that the marginal data likelihood  $p(\mathcal{D}|\mathcal{M})$ , which is used to learn the parameters of the covariance function, might serve as an alternative model quality measure. However, since  $q(\mathcal{D}|\mathcal{M})$  includes the model dimensionality explicitly—which makes it easy to order the search for local models by this criterion—we have used this measure in our experiments.

### 3.2.1. Finding the Network Topology

If no prior knowledge about the body structure of the robot exists, we initialize a fully connected network model containing a total of  $\sum_{k=0}^m \binom{n}{2} \binom{m}{k}$  local models (linking  $m$  actions to  $n$  transformations). Given a set of self observations, the robot can first eliminate 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 (e.g., see Fig. 10). Certain ambiguities will, for instance, remain even after infinitely many training samples. If, for example,  $p_{\mathcal{M}_1}(\mathbf{Z}_{1 \rightarrow 2} | a_1)$  has been determined to be a valid local model, then  $p_{\mathcal{M}_2}(\mathbf{Z}_{1 \rightarrow 2} | a_1, a_2)$  will also be. Although these alternative models might not be distinguishable regarding prediction accuracy for  $\mathbf{Z}_{1 \rightarrow 2}$ , they differ significantly in their complexity and therefore in their model quality  $q(\mathcal{D} | \mathcal{M})$ .

To resolve such locally ambiguous situations and to also find the best topology on a global level, we seek to select the minimal subset  $\mathbb{M} \subset \mathbb{M}_{\text{valid}}$  from the superset of all valid local models  $\mathbb{M}_{\text{valid}} = \{\mathcal{M}_1, \dots\}$  that covers all body part variables and simultaneously maximizes the overall model fit  $q(\mathcal{D} | \mathbb{M}) := \prod_{\mathcal{M} \in \mathbb{M}} q(\mathcal{D} | \mathcal{M})$ .  $\mathbb{M}$  can be found efficiently by computing

the minimal spanning tree of  $\mathbb{M}_{\text{valid}}$  taking the model quality measure of the individual local models as the 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  $|\mathbb{M}| = (n-1)$  local models need to be selected. This yields  $\binom{|\mathbb{M}_{\text{valid}}|}{|\mathbb{M}|}$  possible network structures to be considered. In the typical case, where the robot is composed of 1-DOF joints (arbitrarily connected), 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 like the *observability* of local transformations (from a given camera view point). In practice, straight-forward search heuristics allow us to strongly focus the search on the relevant parts of the structure space, further reducing this number. In our experiments, for instance, we searched by processing the lower dimensional models first. Recall that the quality measure  $q(\mathcal{D} \mid \mathcal{M})$  for a local model is composed of the (data-dependent) prediction accuracy and a (data-independent) complexity penalty. If we consider two valid local models, i.e., with  $\epsilon_{\text{pred}}(\mathcal{D} \mid \mathcal{M}_{1|2}) < \theta$ , then by the definition of  $q(\mathcal{D} \mid \mathcal{M})$ , the quality of a model with lower complexity is always higher compared to a local model with higher complexity for any  $\mathcal{D}$ , i.e.,

$$C(\mathcal{M}_1) < C(\mathcal{M}_2) \iff \forall \mathcal{D} : q(\mathcal{D} \mid \mathcal{M}_1) > q(\mathcal{D} \mid \mathcal{M}_2) . \quad (8)$$

This is due to the fact that  $C(\mathcal{M})$  contains the error threshold  $\theta$  and that this is also the upper bound of all prediction errors  $\epsilon_{\text{pred}}$  (see Eq. 6)—all models above this threshold are invalid and thus discarded. Due to Eq. 8, it

is sufficient to evaluate only the first  $k$  complexity layers of local models in  $\mathbb{M}_{\text{valid}}$  until a minimal spanning tree is found for the first time. This spanning tree then corresponds to the global maximum of overall model quality.

### 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 configuration of the robot for a given action signal (forward modeling) and how to select actions to achieve a given configuration (inverse modeling).

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

$$\begin{aligned} p(\mathbf{X}_1, \dots, \mathbf{X}_n \mid a_1, \dots, a_m) &= \prod_i p(\mathbf{X}_i \mid \text{parents}(\mathbf{X}_i)) \\ &= p(\mathbf{X}_{\text{root}}) \prod_{\mathcal{M}_{i \rightarrow j} \in \mathbb{M}} p_{\mathcal{M}_{i \rightarrow j}}(\Delta_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}) \\ &= p(\mathbf{X}_{\text{root}}) \prod_{\mathcal{M}_{i \rightarrow j} \in \mathbb{M}} p_{\mathcal{M}_{i \rightarrow j}}(\mathbf{X}_i^{-1} \mathbf{X}_j \mid \mathcal{A}_{i \rightarrow j}), \end{aligned} \quad (9)$$

where  $\mathbf{X}_{\text{root}}$  is the position of the robot trunk, which serves as the reference frame for all other body parts. We denoted with  $\mathcal{M}_{i \rightarrow j}$  the local model of  $\mathbb{M}$  which describes the transformation between  $\mathbf{X}_i$  and  $\mathbf{X}_j$ . From  $p(\mathbf{X}_1, \dots, \mathbf{X}_n \mid a_1, \dots, a_m)$  in the factorized form, we can now approximate the maximum likelihood (ML) estimate of the resulting body posture given an action signal by concatenating the geometric transformations of the individual geometric transformations. We may refer the interested reader to (Ware and Lad, 2003) how products of Gaussians can be approximated by a single Gaussian.

Although the *inverse kinematic model* can be derived by applying Bayes’ rule in principle,

$$\begin{aligned} p(a_1, \dots, a_m \mid \mathbf{X}_1, \dots, \mathbf{X}_n) \\ = \frac{p(a_1, \dots, a_m)}{p(\mathbf{X}_1, \dots, \mathbf{X}_n)} p(\mathbf{X}_1, \dots, \mathbf{X}_n \mid a_1, \dots, a_m), \end{aligned}$$

it is difficult in practice to determine the maximum likelihood (ML) solution for the action signal  $a_1, \dots, a_m$ . This is due to the fact that the goal configuration is typically not fully specified for all body parts, but rather for the root part and the end-effector only. Thus, the Bayesian network is constrained at both “ends” only, which results in a high-dimensional optimization problem.

For this reason, we resort to a well-known iterative approach (termed *differential kinematics* in the literature) which applies small changes to the *current* action signal such that the body posture  $\mathbf{X}_1, \dots, \mathbf{X}_n$  approaches the target configuration. Since all individual functions  $f_{\mathcal{M}_i}$  are continuous, and so is the ML posture estimate  $f$  of the forward kinematic model, we can compute the Jacobian of the forward model as

$$J_f(\mathbf{a}) = \left[ \frac{\partial f(\mathbf{a})}{\partial a_1}, \dots, \frac{\partial f(\mathbf{a})}{\partial a_m} \right]^T.$$

Given  $J_f(\mathbf{a})$ , 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 configuration. While such a “greedy” controller may get trapped in local minima of the distance function and fails to plan around obstacles in general, it nevertheless solves many important control tasks and it builds the basis for higher-level path-planning algorithms, such as probabilistic road-maps.



#### 4. Failure Awareness and Life-Long Adaptation

So far, we have assumed that the kinematics of the robot remain unchanged during its whole life-time. It is clear, however, that in real-world applications, the robot will change in the course of time. This requires that the robot revises parts of its internal model over time, allowing it to discriminate between earlier and more recent observations. We would like the robot to detect changes to its body by testing the validity of its local models at different points in time and at different temporal scales. It might even be useful for the robot to maintain multiple body schemas at different time scales. Consider, for example, a robot that uses an accurate pre-programmed model over a long period of time, but which is also able to create and use a new models that takes over as soon as the body structure of the robot changes (which could be as little as the displacement of one visual marker). Such a situation is depicted in Fig. 11. In this experiment, we changed the end-effector body part without notifying the system. The task then was, to automatically detect the change and to learn a replacement for the mismatching local model.

For dealing with model changes over time, we consider *temporal local models*  $\mathcal{M}^T$  that describe the geometric relationship  $p_{\mathcal{M}}^T(\mathbf{Z}_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}, T)$  between two observed body parts  $\mathbf{Y}_i$  and  $\mathbf{Y}_j$  given a subset of the action signal  $\mathcal{A}_{i \rightarrow j} \subset \{a_1, \dots, a_n\}$  and a particular time interval  $T$ . However, the size of the learning problem now also 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 would be practically infeasible to evaluate all of these local models even for small periods of time, three additional assumptions can be made such that

an efficient algorithm for online application can be implemented:

1. Changes to the kinematic structure and/or body geometry are relatively rare events.
2. Changes to the robot’s body 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 Assumption 1, we do not have to re-learn the local models continuously and re-optimize the network, but rather it is sufficient to monitor the prediction accuracies of the models until one of them is not evaluated as being *valid* any more. In this case, Assumption 2 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 Assumption 3, 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.

These rules were incorporated 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*  $C(\mathcal{M}_2 | \mathcal{M}_1)$  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. Experimental Results

We tested our approach in a series of experiments both on a real robot and 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 confidently (blocked joints / deformations),
3. the body schema is updated automatically without human intervention,
4. and 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 6-DOF manipulator, respectively, composed of Schunk PowerCube modules (see Fig. 7). As the *ground-truth* kinematic model to compare against, we consider a carefully hand-tuned model and the joint encoder measurements—which are not used for learning and prediction in our approach. 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, 2004) was used to continuously estimate their 6D poses. The standard deviation of the camera noise was measured to  $\sigma_{\text{markers}} = 44$  mm in 3D space, which is acceptable considering that the camera was located two meters apart from the robot. The prediction errors and errorbars reported in the following were estimated on an independent test set of  $|\mathcal{D}_{\text{testing}}| = 15$  data samples.

### *Evaluation of Model Accuracy*

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 15 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 configuration more accurately than immediate perception. The figure also gives the accuracy of the gradient-based control algorithm. Here, we used an additional marker for defining a target location for the end effector. We learned the full body schema model 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 (approx. 50 mm, see Fig. 15), which is slightly higher than the prediction error alone.

### *Scenario 1: Blocked Joint*

In a second experiment we used the 6-DOF robot (see Fig. 7) to evaluate how well the proposed system can detect a stuck joint and repair its model accordingly. To this aim, we initialized the body schema with an accurate,

manually calibrated model. Upon detection of a model mismatch, new local models were trained from  $|\mathcal{D}_{\text{training}}| = 30$  consecutive training samples recorded after the model was instantiated. In order for a local model to be *valid*, its translational and rotational error on the test set was required to be below a threshold of  $\theta_{\text{trans}} = 3\sigma_{\text{trans}} = 150$  mm and  $\theta_{\text{rot}} = 3\sigma_{\text{rot}} = 45^\circ$ , with  $\sigma_{\text{trans}}$  and  $\sigma_{\text{rot}}$  the standard deviations of the translational and rotational observation noise, respectively. New local models were only sampled when no valid spanning tree could be constructed for  $|\mathcal{D}_{\text{testing}}|$  consecutive time steps, as this is the time it takes to replace most if not all data samples of the test set.

We generated a large sequence of random motor commands  $\langle a_1, \dots, a_m \rangle$ . Before accepting a pose, we checked that the configuration would not cause any (self-)collisions and that the markers of interest would potentially be visible on the camera image. This sequence was sent to the robot and after each motion command, the observed marker positions  $\langle \mathbf{Y}_1, \dots, \mathbf{Y}_n \rangle$  were recorded. We allowed for arbitrary motion patterns (just constrained by the geometry of the manipulator) and thus do not assume 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 time with the initial body layout. In a second run, we blocked the end-effector joint  $a_4$ , 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 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.

Fig. 16 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 newly learned local model quickly increases while the estimated error of the engineered local model decreases rapidly towards its initial accuracy. Later, at  $t = 300$ , the joint gets blocked again in the same position, the accuracy of the previously learned local model increases significantly, and thus the robot can re-use this local model instead of having to learn a new one.

The precision of the combined model—i.e. the engineered one fused with the one learned after having detected the failure—are given in Fig. 17 and Fig. 18 for 20 re-runs of the experiment. The hand-tuned initial geometrical model evaluates to an averaged error at the end-effector of approx. 37 mm. After the joint gets blocked at  $t = 100$ , the error in prediction increases rapidly. After  $t = 115$ , a single new local models gets sampled, which already is enough to bring down the overall error of the combined kinematic model to approximately 51 mm. 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 Tab. 1). At  $t = 300$ , the same quick adaption can be observed when the joint gets blocked again.

### *Scenario 2: Deformed limb*

In a third experiment<sup>1</sup>, we changed the end-effector limb 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 Fig. 11ff). The quantitative results for 20 runs are given in Fig. 20 and Fig. 21. 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 47 mm after recovery. In Tab. 1, we summarize the recovery times for this and the previous experiment. As can be seen from the results, the system adapts to a blocked joint quicker than to a deformed limb, and recalling a previously successful model—i.e. the engineered one after "repair" or the newly learned one after "same failure"—is significantly faster than learning from scratch (after "failure").

---

<sup>1</sup>A demonstration video of this experiment can be found on the Internet at <http://www.informatik.uni-freiburg.de/~sturm>

### *Controlling a Deformed Robot*

Finally, we ran a series of experiments to verify that dynamically maintained body schemas 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 should approach using the inverse kinematics derived from its current body schema, see Fig. 22 and Fig. 23. When the initial geometric model was used to follow the trajectory by using the undamaged manipulator, a positioning accuracy of 7.03mm was measured. When the middle limb was deformed by  $45^\circ$ , the manipulator with a static body schema was significantly off course, leading to an average positioning accuracy of 189.35mm. With dynamic adaptation enabled, the precision settled at 15.24mm. These results are also summarized in Tab. 2 including the two standard deviations of the errors obtained on 20 runs. The results show that dynamic model adaptation enables a robot to maintain a high positioning accuracy even after substantial changes to its body.

## **6. Related Work**

The concept of kinematic functions and inverse kinematic functions in robotics is closely related to the concept of body image and body schema in cognitive neuroscience (Stamenov, 2005; Gallagher, 2005). Evidence from studies of both humans and higher primates indicates regions of the higher-level visual cortex that are specialized for the visual perception of the body (Peelen and Downing, 2007). 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 schema (Holmes and Spence, 2004), suggesting that the body schema serves as a spatio-temporally integrated image of various modalities, such as auditory and visual perceptions and somatic including tactile sensations as well (Sawa et al., 2007). Neuro-physiological evidence indicates that humans as well as higher primates are able to learn and adapt their body schema continuously and autonomously (Meltzoff and Moore, 1997). Brain scan studies of monkeys that have been trained to use tools revealed that the tool itself even gets integrated into their body schemas over time (Maravita and Iriki, 2004).

The problem of learning kinematics of robots has been investigated heavily in the past. Recently, Kolter and Ng (2007) enabled a quadruped robot to follow omnidirectional paths using dimensionality reduction techniques. Their key idea is to use a simulator for identifying a suitable subspace for policies and then to learn with the real robot only in this low-dimensional space. A similar direction has been explored by Dearden and Demiris (2005), who applied dimensionality reduction techniques to unveil the underlying structure of the body schema. Similar to this work, their approach is formulated as a model selection problem between different Bayesian networks. Another instance of approaches based on dimensionality reduction is the work by Grimes et al. (2006) who applied the principal component analysis (PCA) in conjunction with Gaussian process regression for learning walking gaits on a humanoid robot.

Yoshikawa et al. (2004a) used Hebbian networks to discover the body schema from self-occlusion or self-touching sensations. and learned classifiers for body/non-body discrimination from visual data (Yoshikawa et al.,

2004b) . Other approaches used nearest-neighbor interpolation (Morasso and Sanguineti, 1995) or neural networks (Natale, 2004), for example for hand-eye coordination (Gaskett and Cheng, 2003) using self-organizing maps. By combining the body images of multiple modalities, e.g., both a motor and a tactile body image, it becomes possible to infer the motor Jacobians even for invisible hand positions (Sawa et al., 2007). As the required number of training samples increases exponentially with the degrees of freedom of the robot, Lopes and Santos-Victor (2005) propose to learn the kinematic function incrementally, first by moving only the shoulder/elbow joints, and later for the hand. A similar approach was formulated by de Angulo and Torras (2002, 2005) for learning the inverse kinematic function in two parts, which speeds up learning but requires an additional search step during evaluation of the function. For redundant kinematic chains, no global inverse kinematic function exists. Therefore, D’Souza et al. (2001) used Locally Weighted Projection Regression (LWPR) to estimate the inverse kinematic function locally at the current configuration from observed data. Recently, Ting et al. (2006) developed a Bayesian parameter identification method for non-linear dynamic systems, such as a robotic arm or a 7-DOF robotic head.

The approach presented in this article is also related to the problem of self-calibration which can be understood as a subproblem of body schema learning. When the kinematic model is given in a parametric form, the parameters can be estimated efficiently, in certain cases, by optimizing the parameters directly (Hersch et al., 2008) or by maximizing the likelihood of the model given the data (Roy and Thrun, 1999). Genetic algorithms have been used for parameter optimization when no closed form is available

(Bongard et al., 2006b). To a certain extent, such methods can also be used to calibrate a robot that is temporarily using a tool (Nabeshima et al., 2005). In contrast to the work presented here, such approaches require a parametrized kinematic model of the robot.

To achieve continuous self-modeling, Bongard et al. (2006a) recently described a robotic system that continuously learns its own structure from actuation-sensation relationships. In three alternating phases (modeling, testing, prediction), 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, structure learning was studied in arbitrary non-linear systems using similar mechanisms by Bongard and Lipson (2007).

In contrast to all the approaches described above, we propose an algorithm that learns both the structure as well as functional mappings for the individual building blocks. Furthermore, our model is able to revise its structure and component models on-the-fly. This paper presents a complete view on our previous works (Sturm et al., 2008b,a), the first of which introduced the probabilistic kinematic model as well as the basic algorithm for structure search. The latter work extended the model towards life-long adaptation and self-monitoring and gave more experimental results in complex and realistic scenarios.

## 7. Conclusion

In this article, we presented a novel approach to body schema learning and life-long body adaptation for a robotic manipulation system. Our central

idea is to continuously learn a large set of local kinematic models using nonparametric regression and to search for the best arrangement of these models to represent the full system. To the best of our knowledge, this is the first time that such complex kinematic structures have been learned from scratch using visual self-observation only.

Our work follows the general idea of *learning by explanation*. The search for the kinematic structure (including its dimensionality and connectivity) as well as the calibration of the local models is guided by how well the set of observations can be *explained* by the model.

In experiments carried out with a real robot and in simulation, we demonstrated that our system is able to deal with missing and noisy observations, operates in full 3D space, and is able to perform relevant tasks like prediction, control, and online adaptation after failures.

## Acknowledgments

This work has partly been supported by the EU under FP6-IST-004250 (COSY), FP6-IST-045388 (INDIGO), and by the German Ministry for Education and Research (BMBF) through the DESIRE project.

## References

- Bongard, J., Lipson, H., 2007. Automated reverse engineering of nonlinear dynamical systems. In: Proc. of the Nat. Academy of Science. Vol. 104. p. 9943.
- Bongard, J., Zykov, V., Lipson, H., 2006a. Resilient machines through continuous self-modeling. Science 314. (5802), 1118–1121.

- Bongard, J. C., Zykov, V., Lipson, H., 2006b. Automated synthesis of body schema using multiple sensor modalities. In: Proc. of the Int. Conf. on the Simulation and Synthesis of Living Systems (ALIFEX).
- Buss, S. R., su Kim, J., 2005. Selectively damped least squares for inverse kinematics. *Journal of Graphics Tools* 10, 37–49.
- Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., Thrun, S., June 2005. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.
- Craig, J. J., 1989. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- de Angulo, V. R., Torras, C., 2002. Learning inverse kinematics via cross-point function decomposition. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN'02)*. Springer-Verlag, London, UK, pp. 856–864.
- de Angulo, V. R., Torras, C., 2005. Using psoms to learn inverse kinematics through virtual decomposition of the robot. In: *Proceedings of the 8th International Work-Conference on Artificial Neural Networks (IWANN 2005)*. pp. 701–708.
- Dearden, A., Demiris, Y., 2005. Learning forward models for robots. In: *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*. Edinburgh, Scotland, pp. 1440–1445.
- D'Souza, A., Vijayakumar, S., Schaal, S., 2001. Learning inverse kinematics.

- In: Proceedings of the Intelligent Robots and Systems (IROS 2001). Vol. 1. pp. 298–303 vol.1.
- Fiala, M., 2004. Artag, a fiducial marker system using digital techniques. Tech. rep., National Research Council Canada.
- Gallagher, S., 2005. How the Body Shapes the Mind. Clarendon Press, Oxford, UK.
- Gaskett, C., Cheng, G., 2003. Online learning of a motor map for humanoid robot reaching. In: Proceedings of the 2nd International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003).
- Grimes, D., Chalodhorn, R., Rao, R., 2006. Dynamic imitation in a humanoid robot through nonparametric probabilistic inference. In: Proc. of the Robotics: Science and Systems (RSS). Philadelphia, USA.
- Hersch, M., Sauser, E., Billard, A., 2008. Online learning of the body schema. *International Journal of Humanoid Robotics* 5 (2), 161–181.
- Holmes, N. P., Spence, C., 2004. The body schema and the multisensory representation(s) of peripersonal space. *Cognitive processes* 5 (2), 94–105.
- Jensen, F. V., 2001. Bayesian Networks and Decision Graphs. Springer, New York, NY.
- Kolter, J., Ng, A., June 2007. Learning omnidirectional path following using dimensionality reduction. In: Proceedings of Robotics: Science and Systems. Atlanta, GA, USA.

- Lopes, M., Santos-Victor, J., June 2005. Visual learning by imitation with motor representations. *IEEE Transactions on Systems, Man, and Cybernetics* 35 (3), 438–449.
- Maravita, A., Iriki, A., 2004. Tools for the body (schema). *Trends in Cognitive Sciences* 8 (2), 79–86.
- Meltzoff, A. N., Moore, M. K., 1997. Explaining facial imitation: A theoretical model. *Early Development and Parenting* 6, 179–192.
- Morasso, P., Sanguineti, V., 1995. Self-organizing body-schema for motor planning. *Journal of Motor Behavior* 26, 131–148.
- Nabeshima, C., Kuniyoshi, Y., Lungarella, M., 2006. Adaptive body schema for robotic tool-use. *Journal of Advanced Robotics* 10 (20), 1105–1126.
- Nabeshima, C., Lungarella, M., Kuniyoshi, Y., 2005. 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)*. pp. 7–12.
- Natale, L., May 2004. Linking action to perception in a humanoid robot: A developmental approach to grasping. Ph.D. thesis, University of Genoa, Italy.
- Pearl, J., 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

- Peelen, M. V., Downing, P. E., 2007. The neural basis of visual body perception. *Nature Reviews Neuroscience* 8 (8), 636–648.
- Rasmussen, C. E., Williams, C. K., 2006. *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge, Massachusetts.
- Rosales, E. M., Gan, Q., 2004. Forward and inverse kinematics models for a 5-dof pioneer 2 robot arm. Tech. Rep. CSM-413, University of Essex, UK.
- Roy, N., Thrun, S., 1999. Online self-calibration for mobile robots. In: *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*.
- Sawa, F., Ogino, M., Asada, M., 2007. Body image constructed from motor and tactile images with visual information. *International Journal of Humanoid Robotics* 4 (2), 347–364.
- Sciavicco, L., Siciliano, B., January 2000. *Modelling and Control of Robot Manipulators*. Springer.
- Stachniss, C., Hähnel, D., Burgard, W., Grisetti, G., 2005. On actively closing loops in grid-based FastSLAM. *Advanced Robotics* 19 (10), 1059–1080.
- Stamenov, M. I., 2005. *Body Image and Body Schema: Interdisciplinary Perspectives on the Body*. John Benjamins Publishing Co, Ch. 2.
- Sturm, J., Plagemann, C., Burgard, W., 2008a. Adaptive body scheme models for robust robotic manipulation. In: *Proc. of the Robotics: Science and Systems (RSS)*. Zürich, Switzerland.



- Sturm, J., Plagemann, C., Burgard, W., 2008b. Unsupervised body scheme learning through self-perception. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). Pasadena, CA, USA.
- Sudderth, E. B., Ihler, A. T., Freeman, W. T., Willsky, A. S., 2003. Nonparametric belief propagation. In: Computer Vision and Pattern Recognition.
- Ting, J., Mistry, M., Peters, J., Schaal, S., Nakanishi, J., August 2006. A bayesian approach to nonlinear parameter identification for rigid body dynamics. In: Proceedings of Robotics: Science and Systems. Philadelphia, USA.
- Ware, R., Lad, F., 2003. Approximating the distribution for sums of products of normal variables. Tech. Rep. UCDMS 2003/15, University of Canterbury, England.
- Yoshikawa, Y., Hosoda, K., Asada, M., 2004a. 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, pp. 135–138.
- Yoshikawa, Y., Tsuji, Y., Hosoda, K., Asada, M., 2004b. 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.

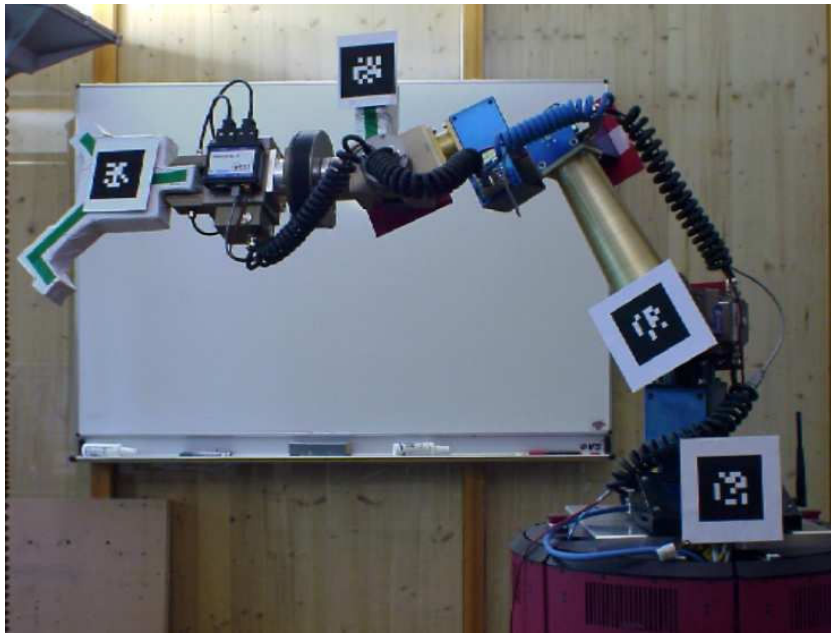


Figure 1: Our 6-DOF robotic manipulator arm learns and monitors its own body schema using an external monocular camera and visual markers.

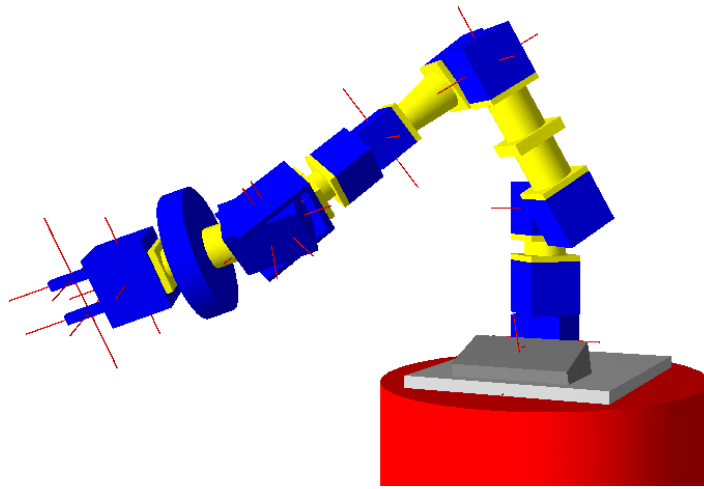


Figure 2: A similar, simulated 7-DOF-manipulator consisting of 10 body parts.

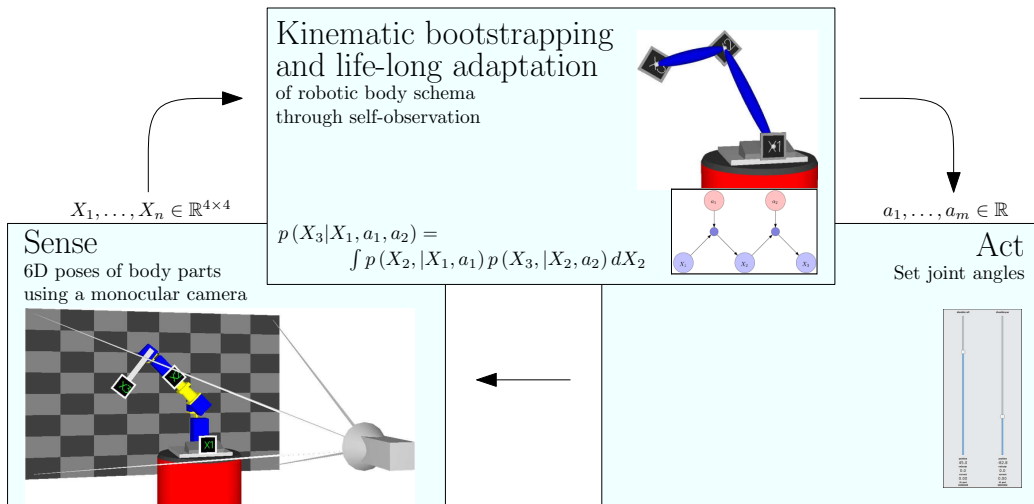


Figure 3: Learning the kinematic function from self-observations. Lower right: The robot sends action commands to its motors. Lower left: It senses the new configuration of its body parts using a camera. Top: The robot learns and continuously updates a probabilistic kinematic model from these action-effect samples.

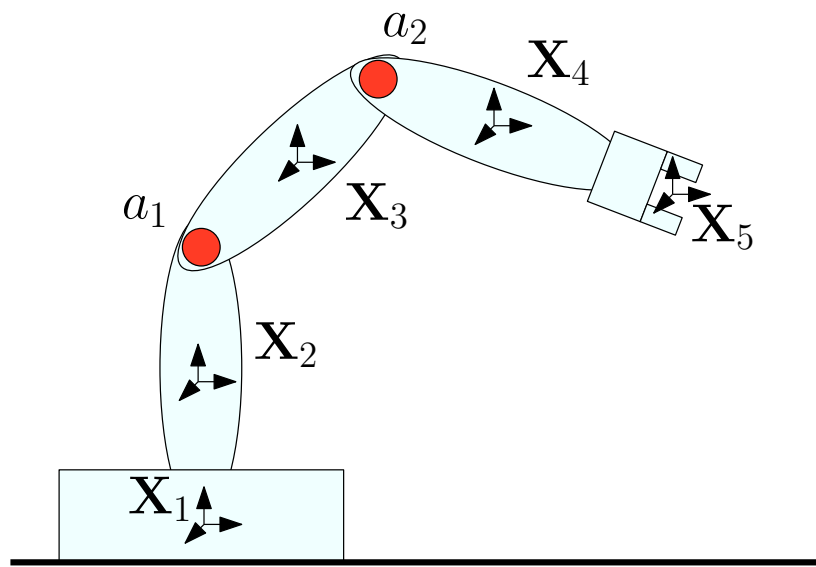


Figure 4: The kinematic function of a robotic manipulator is constructed by concatenating the individual geometric transformations of each of the joints and links.

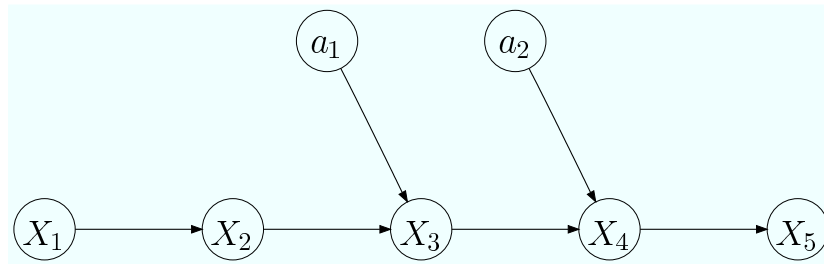


Figure 5: The kinematic function of the robot depicted in Fig. 4 represented as a Bayesian network. An edge indicates a probabilistic dependency, while the absence of an edge indicates probabilistic independency.

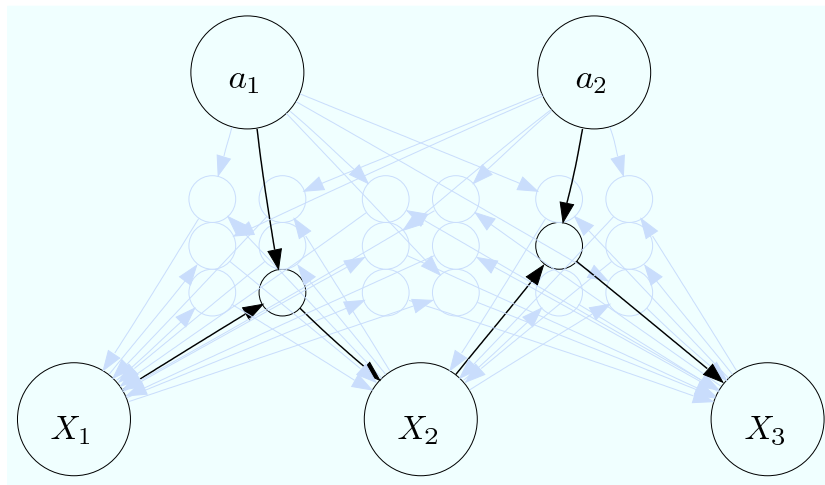


Figure 6: Example network model linking action signals  $a_1$  and  $a_2$  to body part locations  $X_1$ ,  $X_2$ , and  $X_3$  using local models (green circles). The black edges indicate the current best minimal spanning tree found during the search for the most compact model.

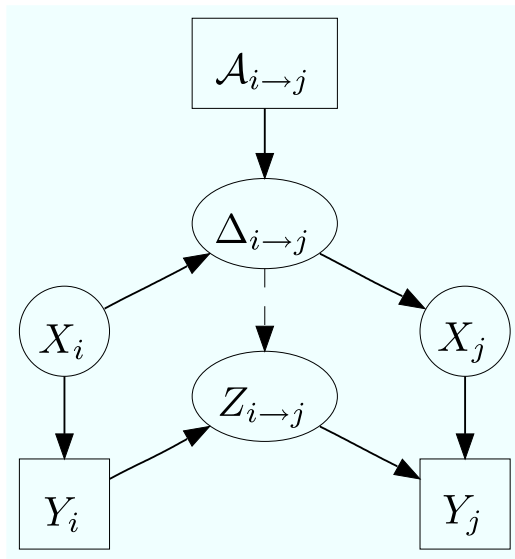


Figure 7: Local model for two body parts  $\mathbf{X}_i$  and  $\mathbf{X}_j$  as well as their dependent variables.  $\mathcal{A}_{i \rightarrow j}$  denotes the set of independent action variables that cause a local transformation  $\Delta_{i \rightarrow j}$ .  $\mathbf{Y}_i$  and  $\mathbf{Y}_j$  are the observed part locations, and  $\mathbf{Z}_{i \rightarrow j}$  is their relative geometric transformation.



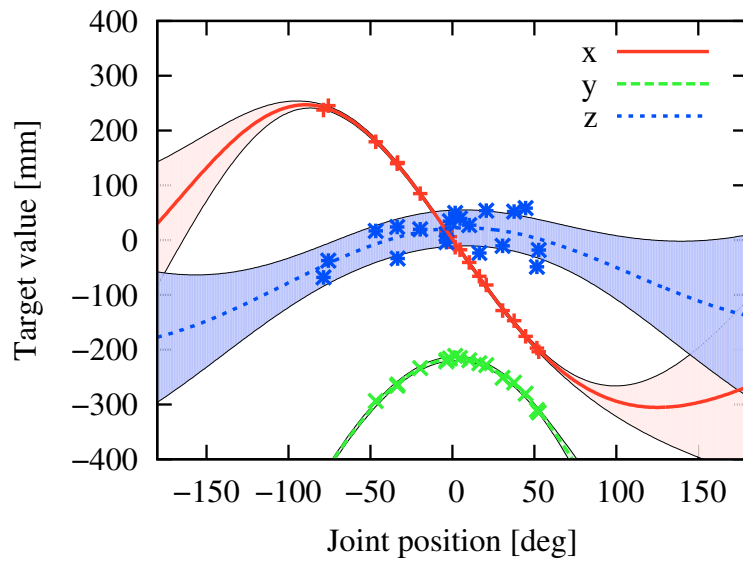


Figure 8: Example of an accurate local model learned for two body parts and an action variable. Note the low predictive variance for the  $x$ - and  $y$  components as well as the higher noise in the  $z$  dimension, which is due to higher measurement uncertainty in this direction.

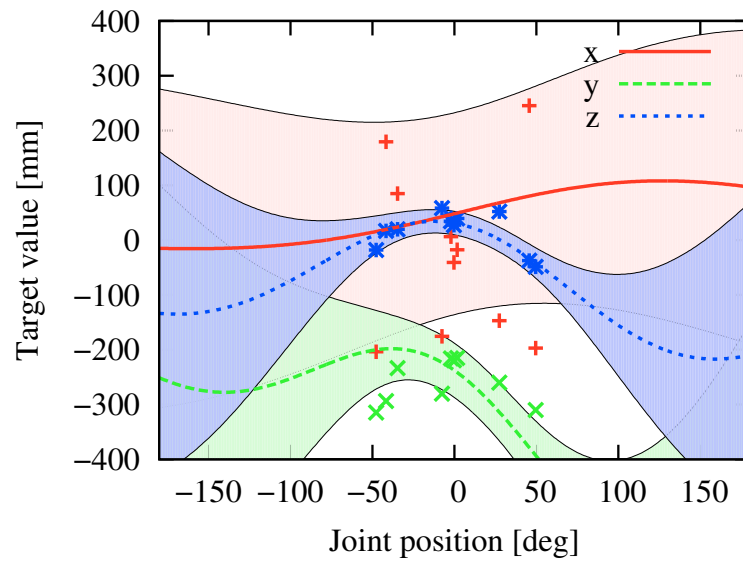


Figure 9: Less accurate model learned for the same body parts as in Fig. 8 but for a different action variable. Such a local model is less likely to be part of the Bayesian network describing the full kinematic chain of the robot since, on average, its predictions are less accurate.

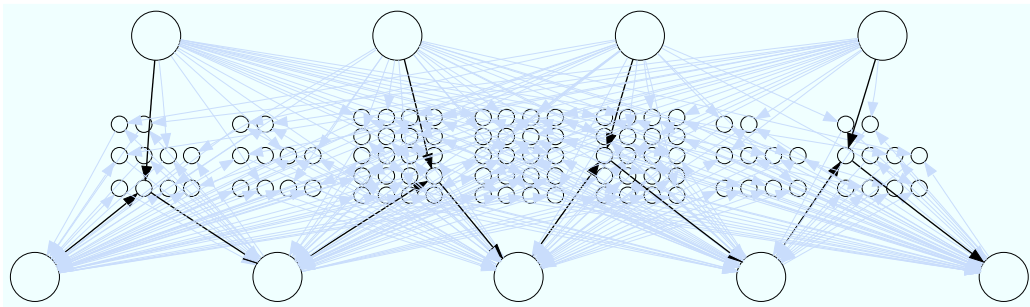


Figure 10: In an early learning phase, the robot knows only little about its body structure and, thus, all possible local models need to be considered in parallel (gray edges). Using the subset of valid local models, a minimal spanning tree can be constructed to form a sparse Bayesian network (dark black edges). This can subsequently be used as a body schema for prediction and control.

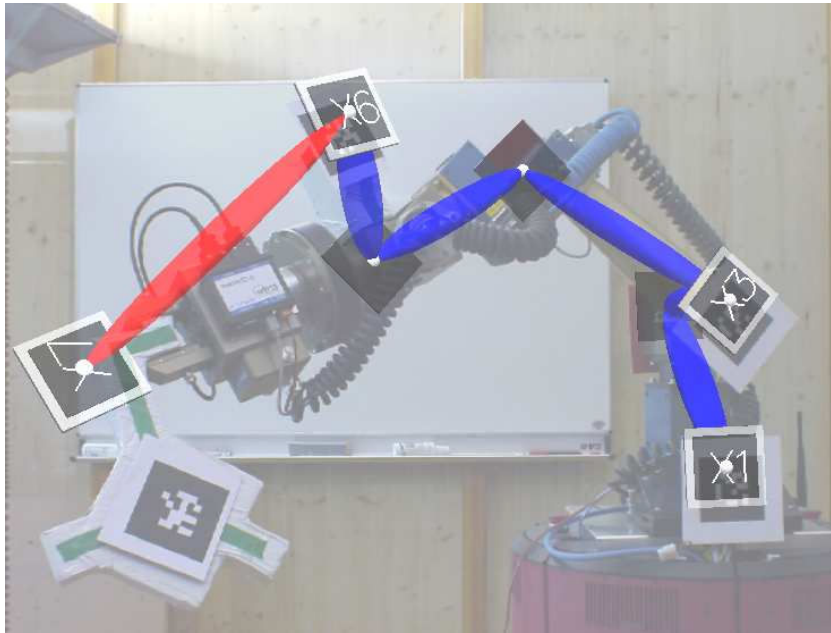


Figure 11: After a different tool is placed in the gripper, the model predictions do not fit the current observations anymore.

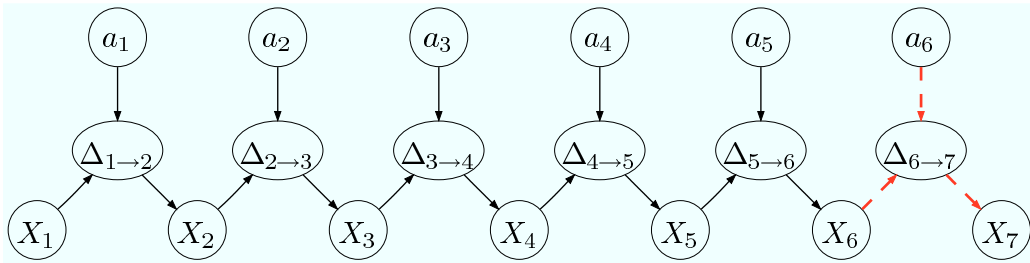


Figure 12: Continued experiment from Fig. 11. The current body schema linking action signals  $a_i$  and body parts  $\mathbf{X}_j$  using local models  $\Delta_{j \rightarrow k}$ . Here, a mismatch between the internal model and recent self-observation has been detected at node  $\Delta_{6 \rightarrow 7}$ .

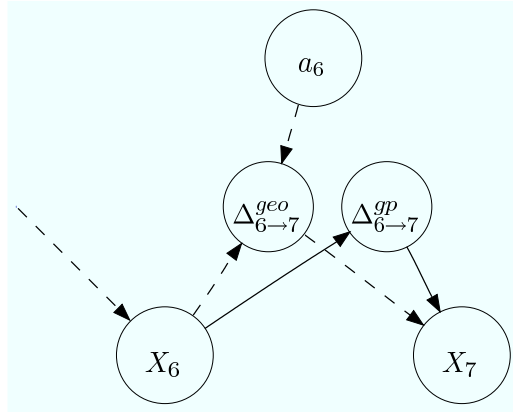
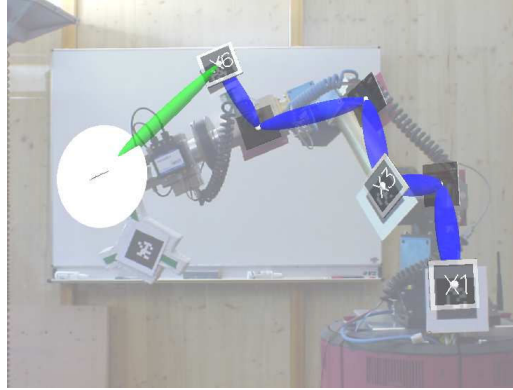


Figure 13: Continued experiment from Fig. 12. The robot samples a local model as replacement for the mismatching component  $\Delta_{6 \rightarrow 7}$ . The first newly sampled model ( $\Delta_{6 \rightarrow 7}^{gp}$ ) has high uncertainty, because of the missing dependency on action  $a_6$ .

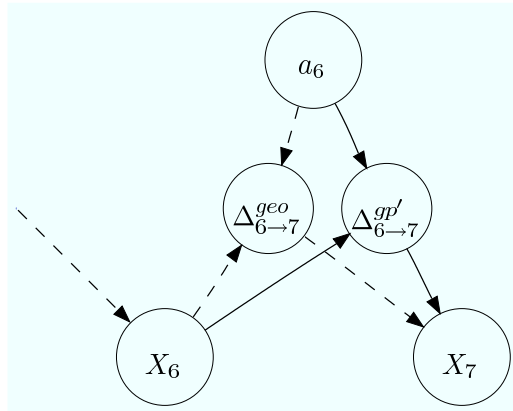
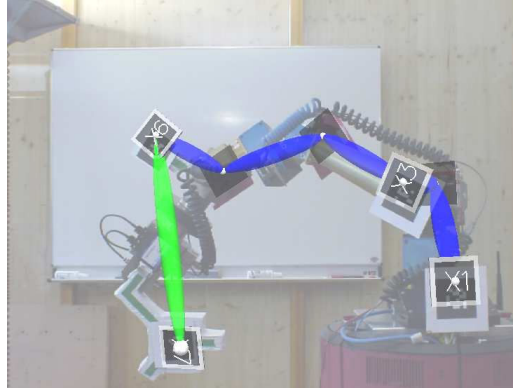


Figure 14: Continued experiment from Fig. 13. The second sampled model ( $\Delta_{6 \rightarrow 7}^{gp'}$ ) is a more suitable replacement for the mismatching component.

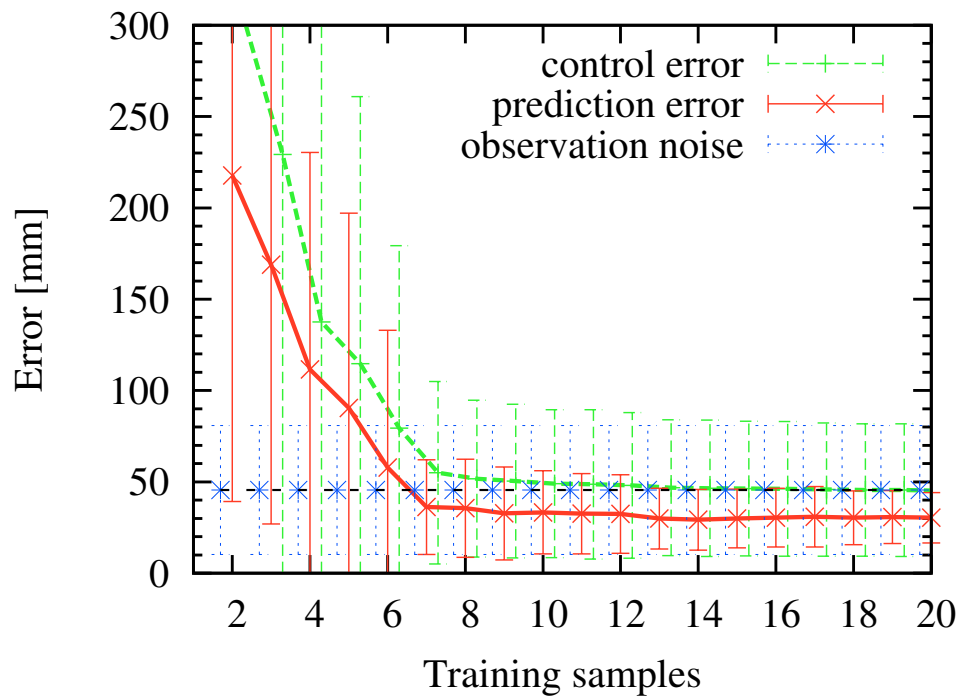


Figure 15: Prediction and control errors for a kinematic model that is learned from scratch. Already after 7 samples, the average prediction error is lower than the average localization error of the visual markers.



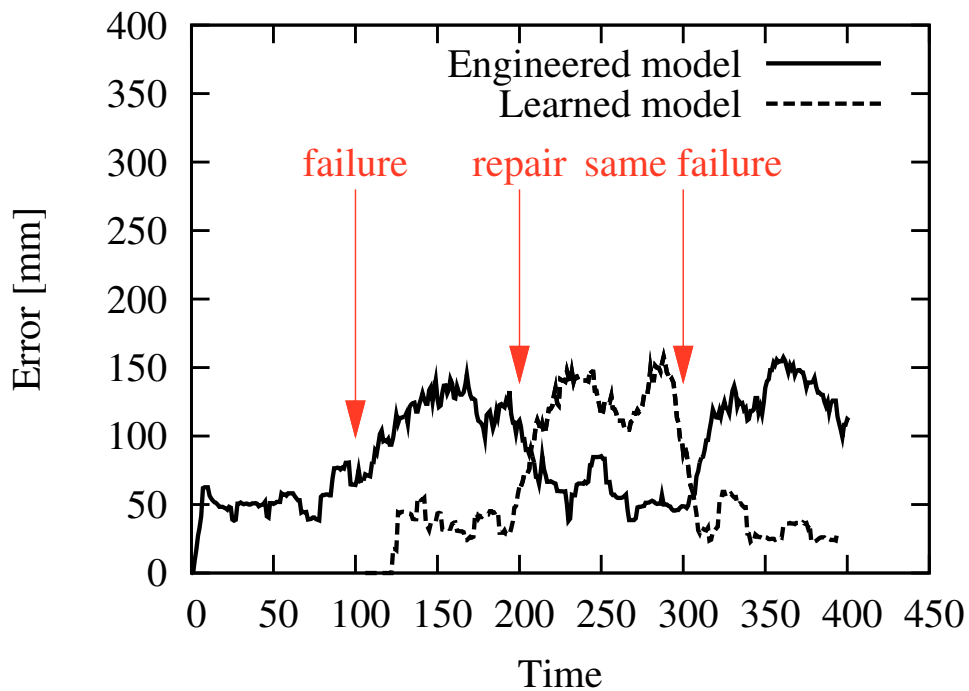


Figure 16: At  $t = 100$ , a joint gets blocked, which causes the initial local model  $p_{\text{engineered}}(\mathbf{Z}_{6 \rightarrow 7} \mid a_4)$  to produce substantially larger prediction errors. At  $t = 126$ , the robot samples a new local model  $p_{\text{learned}}(\Delta_{6 \rightarrow 7})$  as replacement.

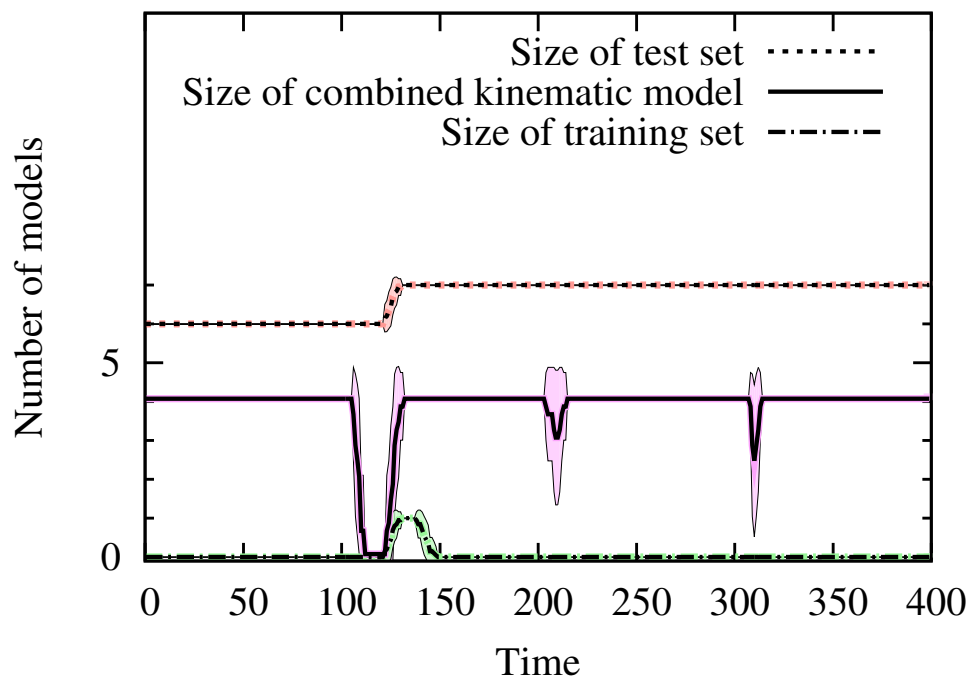


Figure 17: Evolution of the number of models in time, averaged over 20 runs. When the joint is blocked at  $t = 100$ , the combined kinematic model becomes invalid. A new model is instantiated and added to the training and test sets.

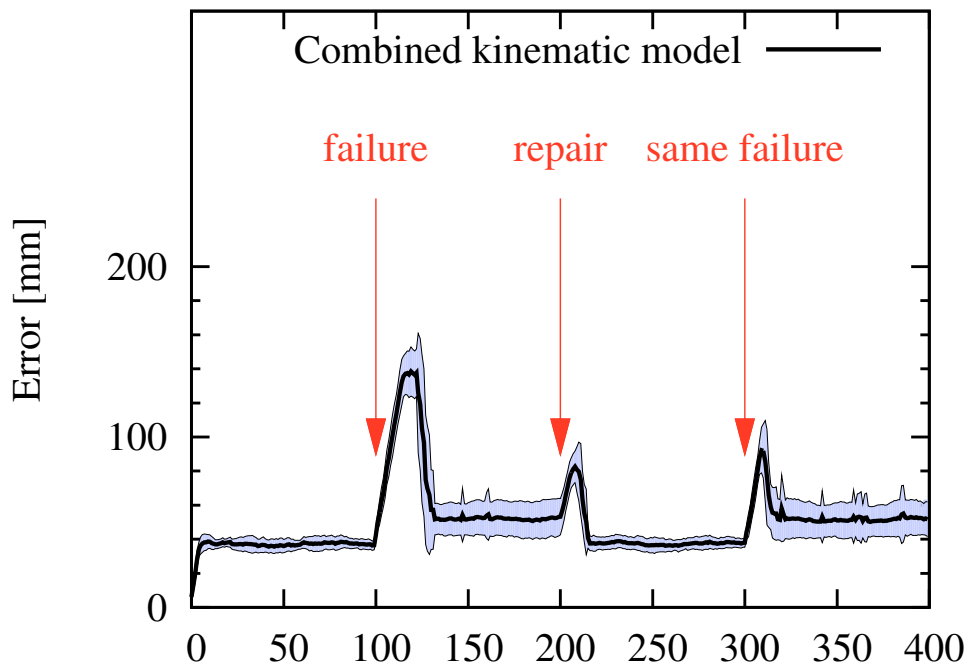


Figure 18: The absolute prediction error of the combined kinematic model  $p(\mathbf{Z}_{1 \rightarrow 7} \mid a_1, \dots, a_6)$  of our 6-DOF manipulator averaged over 20 runs. As can be seen from the plot, the prediction accuracy recovers quickly after each of the three external events.

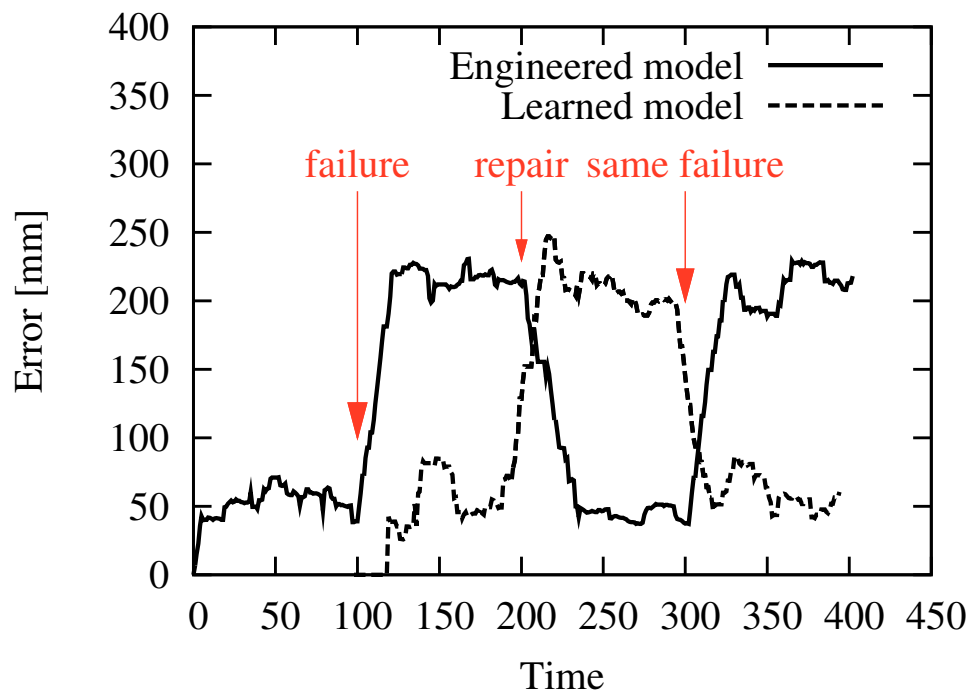


Figure 19: In this experiment, the limb length and orientation of the end-effector were changed at  $t = 100$ .

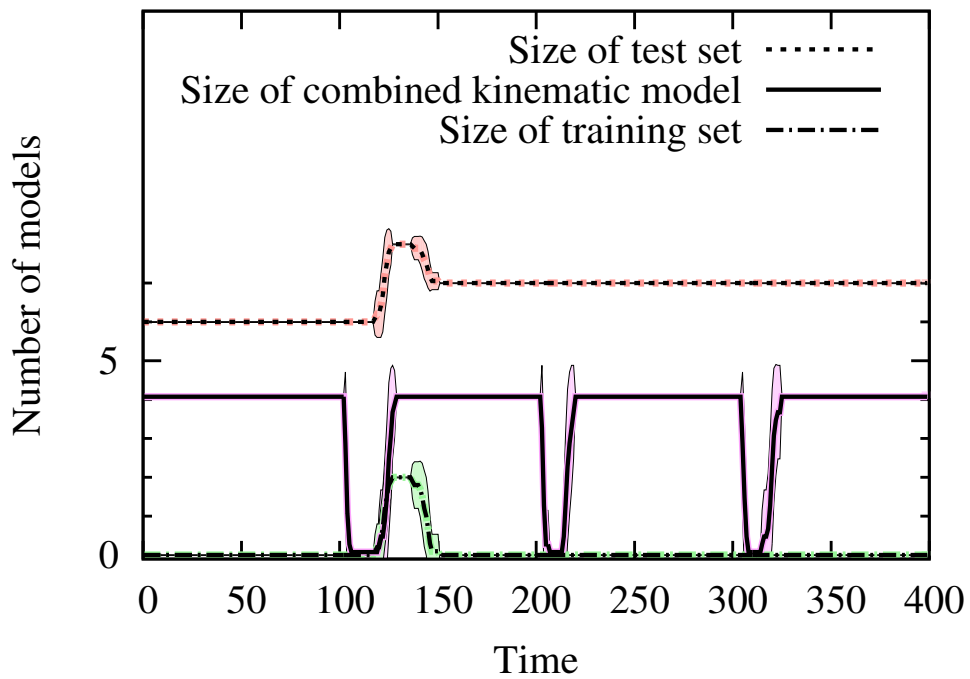


Figure 20: Evolution of the number of models in time for the experiment described in Fig. 19, averaged over 20 runs. The robot had to sample on average two local models before the combined kinematic model could be restored.

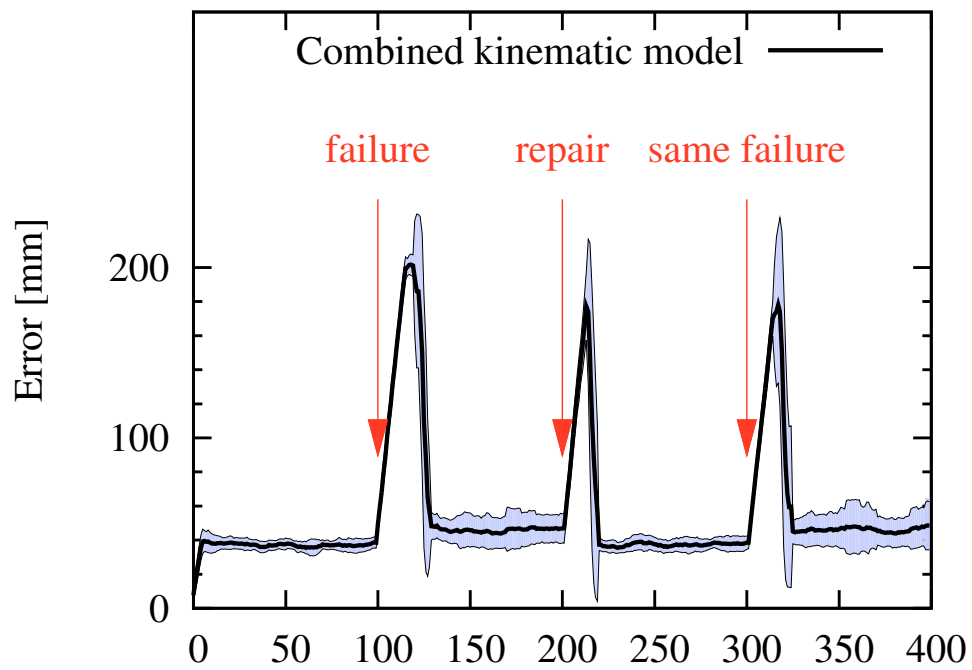


Figure 21: Evaluation of the prediction accuracy of the experiment in Fig. 19, averaged over 20 runs. The robot is able to recover quickly from the large prediction errors induced by the limb deformation.

Table 1: Evaluation of the recovery time required after being exposed to different types of failures. We give the mean recovery times for 20 runs, respectively, and the ( $\pm$ ) standard deviation spread.

Visibility rate	Failure type	Recovery time after		
		failure	repair	same failure
91.9%	Joint blocked	16.50 $\pm$ 1.20	0.45 $\pm$ 0.86	0.65 $\pm$ 1.15
79.0%	Limb deformed	20.20 $\pm$ 1.96	11.10 $\pm$ 0.83	12.10 $\pm$ 1.64

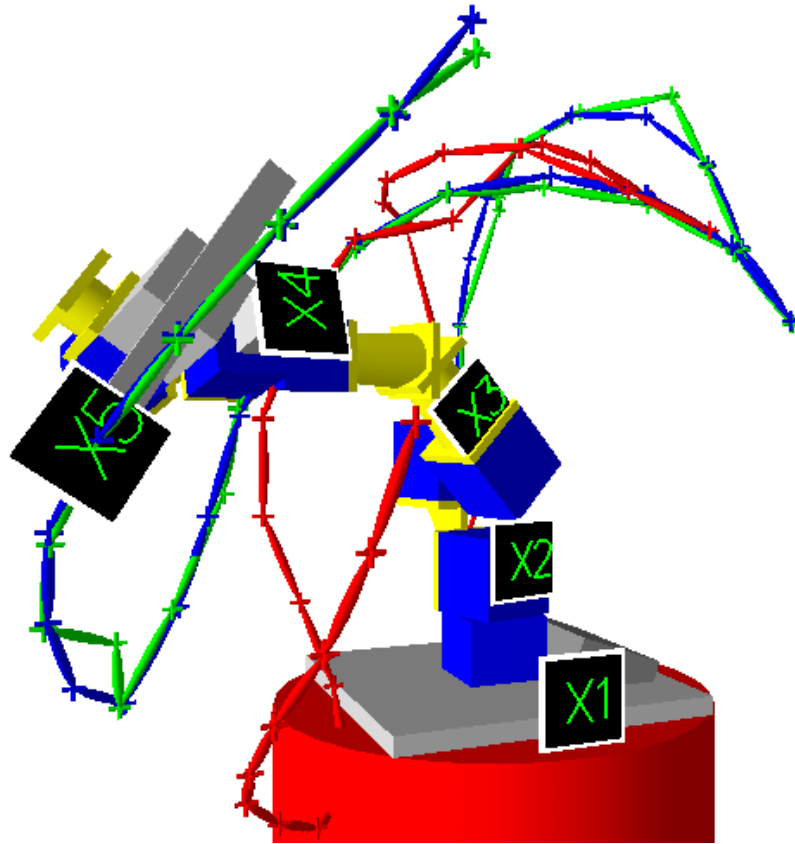


Figure 22: The manipulator robot with a deformed limb has to follow the blue target trajectory. With a static body model, it suffers from strong deviation (red trajectory). By using our approach, the body schema is dynamically adapted, and the trajectory is very well approached (green trajectory).



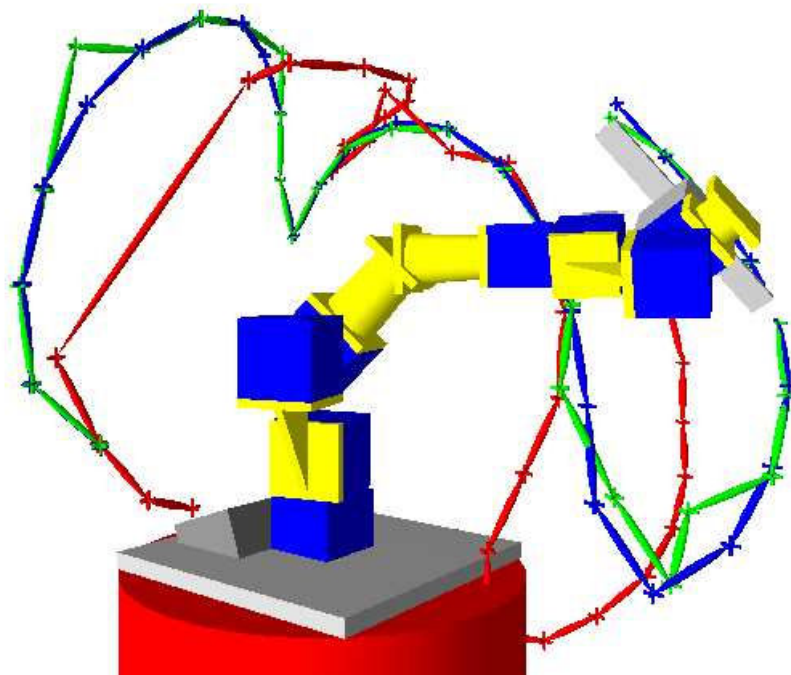


Figure 23: Same as Fig. 22, but here from a different perspective.

Table 2: Evaluation of the inverse kinematics experiment in simulation. By sampling new local models upon the detection of a failures, the robot is able to regain high positioning accuracy.

Body shape	Model type	Control error
initial (unchanged)	static	$7.03 \pm 10.87$
deformed	static	$189.35 \pm 28.36$
deformed	adaptive	$15.24 \pm 1.86$