

Adaptive Body Scheme Models for Robust Robotic Manipulation

Jürgen Sturm Christian Plagemann Wolfram Burgard

Albert-Ludwigs-University Freiburg, Department for Computer Science, 79110 Freiburg, Germany
 {sturm, plagem, burgard}@informatik.uni-freiburg.de

Abstract—Truly autonomous systems require the ability to monitor and adapt their internal body scheme throughout their entire lifetime. In this paper, we present an approach allowing a robot to learn from scratch and maintain a generative model of its own physical body through self-observation with a single monocular camera. We represent the robot’s internal model as a compact Bayesian network, consisting of local models that describe the physical relationships between neighboring body parts. We introduce a flexible Bayesian framework that allows to simultaneously select the maximum-likely network structure and to learn the underlying conditional density functions. Changes in the robot’s physiology can be detected by identifying mismatches between model predictions and the self-perception. To quickly adapt the model to changed situations, we developed an efficient search heuristic that starts from the structure of the best explaining memorized network and then replaces local components where necessary. In experiments carried out with a real robot equipped with a 6-DOF manipulator as well as in simulation, we show that our system can quickly adapt to changes of the body physiology in full 3D space, in particular with limited visibility, noisy and partially missing observations, and without the need for proprioception.

I. INTRODUCTION

Autonomous robots deployed in real world environments have to deal with situations in which components change their behavior or properties over time. Such changes can for example come from deformations of robot parts or material fatigue. Additionally, to make proper use of tools, a robot should be able to incorporate the tool into its own body scheme and to adapt the gained knowledge in situations in which the tool is grabbed differently. Finally, components of the robot might get exchanged or replaced by newer parts that no longer comply with the models engineered originally.

Kinematic models are widely used in practice, especially in the context of robotic manipulation [1, 2]. These models are generally derived analytically by an engineer [3] and usually rely heavily on prior knowledge about the robots’ geometry and kinematic parameters. As robotic systems become more complex and versatile or are even delivered in a completely reconfigurable way, there is a growing demand for techniques allowing a robot to automatically learn body schemes with no or only minimal human intervention.

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 body scheme changes, e.g., through regular wear-and-tear over time. Furthermore, the

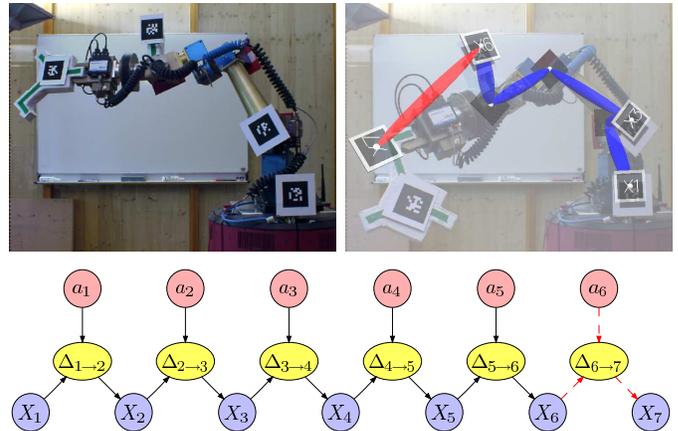


Fig. 1. **Upper left:** Our 6-DOF robotic manipulator arm learns and monitors its own body-scheme using an external monocular camera and visual markers. **Upper right:** After a different tool is placed in the robot’s end-effector, the model predictions do not fit the current observations anymore. **Bottom:** The current body scheme linking action signals a_i and body parts X_j using local models $\Delta_{j \rightarrow k}$. Here, a mismatch between the internal model and recent self-observation has been detected at $\Delta_{6 \rightarrow 7}$.

ability to learn a body scheme is important in the context of tool use scenarios in which a robot has to identify the effects of its actions on the tool.

In this paper, we investigate how to equip autonomous robots with the ability to learn and adapt their own body schemes and kinematic models using exploratory actions and self-perception only. We propose an approach to learn a Bayesian network for the robot’s kinematic structure including the forward and inverse models relating action commands and body pose. More precisely, we start with a fully connected network containing all perceivable body parts and available action signals, perform random “motor babbling,” and iteratively reduce the network complexity by analyzing the perceived body motion. At the same time, we learn non-parametric regression models for all dependencies in the network, which can later be used to predict the body pose when no perception is available or to allow for gradient-based posture control.

One of the major advantages of the approach presented in this paper is that it addresses all of the following practical problems that frequently arise in robotic manipulation tasks in a single framework:

- **Prediction:** If both the structure and the CDFs of the

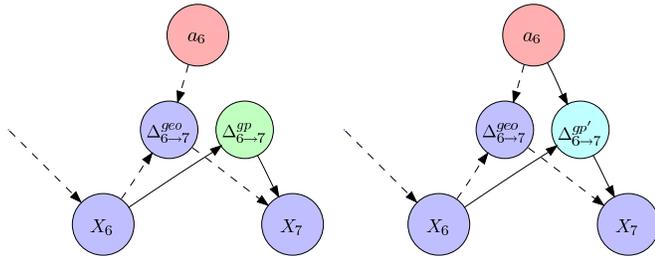
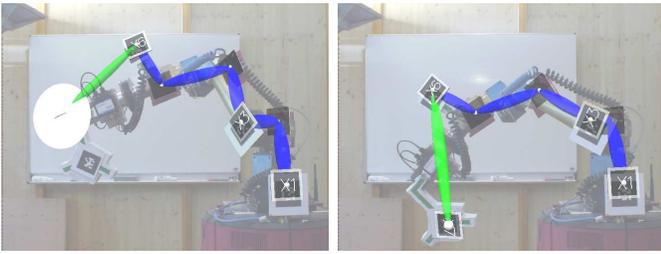


Fig. 2. Continued experiment from Figure 1. The robot samples a local model as replacement for the mismatching component $\Delta_{6 \rightarrow 7}$. **Left:** The first newly sampled model ($\Delta_{6 \rightarrow 7}^{gp}$) has high uncertainty, because of the missing dependency on action a_6 . **Right:** The second sampled model ($\Delta_{6 \rightarrow 7}^{gp'}$) is a more suitable replacement for the mismatching component.

Bayesian network are known, the robot is able to predict for a given action command the expected resulting body configuration.

- **Control:** Conversely, given a target body pose, our approach is able to generate appropriate action commands that will lead to this pose.
- **Model testing:** Given both a prediction and an observation of the current body pose, the robot is able to estimate the accuracy of its own pose predictions. Model accuracy can, for example, be defined in terms of a distance metric or a likelihood function.
- **Learning:** Given a sequence of action signals and the corresponding body postures, the Bayesian network and its parameters can be learned from the data.
- **Discovering the network structure:** When the structure of the Bayesian network is unknown, the robot is able to build it from the available local models which are most consistent with the observed data.
- **Failure detection and model adaptation:** When the robot's physiology changes, e.g., when a joint gets blocked or is deformed, or a visual marker is changed, this is efficiently detected so that only the affected local models of the Bayesian network need to be replaced.

II. RELATED WORK

The problem of learning kinematics of robots has been investigated heavily in the past. For example, Kolter and Ng [4] enable a quadruped robot to learn how to follow omnidirectional paths using dimensionality reduction techniques and based on simulations. Their key idea is to use the 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 *et al.* [5], who applied dimensionality reduction techniques to unveil the underlying structure of the body scheme. 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.* [6] who applied the principal component analysis (PCA) in conjunction with Gaussian process regression for learning walking gaits on a humanoid robot.

In previous work [7], we have presented an approach to deal with the problem of learning a probabilistic self-model for a robotic manipulator. This approach, however, neither covered aspects of failure detection and life-long model revision nor did it address partial observability of model components. In this work, we give a more rigorous formulation of the body-scheme learning framework, we significantly extend the model toward life-long adaptation and self monitoring, and we give experimental results in complex and realistic scenarios.

Yoshikawa *et al.* [8] used Hebbian networks to discover the body scheme from self-occlusion or self-touching sensations. Later, [9] learned classifiers for body/non-body discrimination from visual data. Other approaches used for example nearest-neighbor interpolation [10] or neural networks [11]. Recently, Ting *et al.* [12] developed a Bayesian parameter identification method for nonlinear dynamic systems, such as a robotic arm or a 7-DOF robotic head.

The approach presented in this paper is also related to the problem of self-calibration which can be understood as a subproblem of body scheme learning. When the kinematic model is known up to some parameters, they can in certain cases be efficiently estimated by maximizing the likelihood of the model given the data [13]. Genetic algorithms have been used by Bongard *et al.* [14] for parameter optimization when no closed form is available. To a certain extent, such methods can also be used to calibrate a robot that is temporarily using a tool [15]. In contrast to the work presented here, such approaches require a parameterized kinematic model of the robot.

To achieve continuous self-modeling, Bongard *et al.* [16] 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, Bongard *et al.* [17] studied structure learning in arbitrary non-linear systems using similar mechanisms.

In contrast to all the approaches described above, we propose an algorithm that both learns 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.

III. A BAYESIAN FRAMEWORK FOR ROBOTIC BODY SCHEMES

A robotic body scheme describes the relationship between available action signals $\langle a_1, \dots, a_m \rangle$, self-observations $\langle Y_1, \dots, Y_n \rangle$, and the configurations of the robot's body parts $\langle X_1, \dots, X_n \rangle$. In our concrete scenario, in which we consider the body scheme 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 joint angles. Whereas the $X_i \in \mathbb{R}^6$ encode the 6-dimensional poses (3D Cartesian position and 3D Euler angles) of the body parts w.r.t. a reference coordinate frame, the $Y_i \in \mathbb{R}^6$ are generally noisy and potentially missing observations of the body parts. Throughout this paper, we use capital letters to denote 6D pose variables to highlight that these also uniquely define homogeneous transformation matrices, which can be concatenated and inverted. Note that we do not assume direct feedback/proprioception telling the robot how well joint i has approached the requested target angle a_i .

Formally, we seek to learn the probability distribution

$$p(X_1, \dots, X_n, Y_1, \dots, Y_n \mid a_1, \dots, a_m), \quad (1)$$

which in this form is intractable for all but the simplest scenarios. To simplify the problem, it is typically assumed that each observation variable Y_i is independent from all other variables given the true configuration X_i of the corresponding body part and that they can thus be fully characterized by an observation model $p(Y_i \mid X_i)$. 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

$$p(X_1, \dots, X_n \mid a_1, \dots, a_m) = \prod_i p(X_i \mid \text{parents}(X_i)) \cdot p(\text{parents}(X_i) \mid a_1, \dots, a_m). \quad (2)$$

Here, $\text{parents}(X_i)$ denotes the set of locations of body parts, which are directly connected to body part i .

The main idea behind this work is to make the factorized structure of the problem explicit by introducing (hidden) transformation variables $\Delta_{i \rightarrow j} := X_i^{-1} X_j$ for all pairs of body parts (X_i, X_j) as well as their observed counterparts $Z_{i \rightarrow j} := Y_i^{-1} Y_j$. Here, we use the 6D pose vectors X and Y as their equivalent homogeneous transformation matrices, which means that $\Delta_{i \rightarrow j}$ reflects the (deterministic) relative transformation between body parts X_i and X_j . Figure 3 depicts a *local model*, which fully defines the relationship between any two body parts X_i and X_j and their dependent variables, if all other body parts are ignored.

Since local models are easily invertible ($\Delta_{i \rightarrow j}$ are homogeneous transformations), any set of $n - 1$ local models which form a spanning tree over all n body parts defines a model for the whole kinematic structure.

In the following, we explain (1) how to continuously learn local models from data and (2) how to find the best spanning

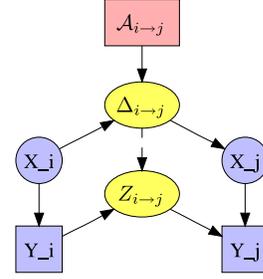


Fig. 3. Graphical model for two body parts X_i and X_j as well as their dependent variables. \mathcal{A} denotes the set of independent action variables that cause a local transformation $\Delta_{i \rightarrow j}$. Y_i and Y_j are the observed part locations, and $Z_{i \rightarrow j}$ is their relative geometric transformation.

tree built from these local models that explains the whole robot. In this work, we consider the single best solution only and do not perform model averaging over possible alternative structures.

Please note that in theory, it would be straight-forward to keep multiple structure hypotheses and to average over them using Bayes' rule. Control under structure uncertainty is a slightly more difficult problem. One would have to average over all possible structures and assess the individual risks and gains for possible actions. Then, the one action sequence should be selected that maximizes the overall gain while keeping all possible risks low [18].

In practice, we found that considering the most-likely structure only is sufficient for most relevant tasks. Our approach is conservative in this respect since it requires a certain minimal model accuracy from all parts of the body scheme.

A. Local Models

The local kinematic models are the central concept in our body scheme framework. A *local model* \mathcal{M} describes the geometric relationship $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j})$ between two observed body parts Y_i and Y_j , given a subset of the action signal $\mathcal{A}_{i \rightarrow j} \subset \{a_1, \dots, a_n\}$.

The probability distribution underlying a local model can be defined in various ways. If an analytic model of the robot exists from its specifications, it can be used directly to construct $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j})$. The standard way to describe a geometric model for robot manipulators is in terms of the Denavit-Hartenberg parameters [1, 19]. When available, the advantages of these models are outstanding: they are exact and efficient in evaluation. In practice, however, such models need to be calibrated carefully and often require re-calibration after periods of use.

B. Learning Local Models from Noisy Observations

On the real robotic platform used in our experiments, the actions a_i correspond to the target angle requested from joint i and the observations Y_i are obtained by tracking visual markers in 3D space including their 3D orientation [20] (see the top right image of Figure 1). Note that the Y_i 's are inherently noisy and that missing observations are common, for example in the case of (self-)occlusion.

The probability distribution $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j})$ of a local model \mathcal{M} can be learned from a sequence of observations $\mathcal{D} = \{(Z_{i \rightarrow j}^t, \mathcal{A}_{i \rightarrow j}^t)\}_{t=1}^t$. If we assume Gaussian white noise with zero mean on the observations, the sensor model becomes $Y_i \sim X_i + \mathcal{N}(0, \sigma_{\text{sensor}})$. Note that we can connect the two body parts X_i and X_j in Figure 3 either by learning $p_{\mathcal{M}}(\Delta_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j})$ or $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j})$. The link $p(\Delta_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}) = p(X_i^{-1} X_j \mid \mathcal{A}_{i \rightarrow j})$ is noise-free. It, however, requires inference starting from Y_i and Y_j through both observation models via the indirect Bayesian pathway $Y_i \leftarrow X_i \rightarrow \Delta_{i \rightarrow j} \rightarrow X_j \rightarrow Y_j$. Thus, we propose to learn the model for $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}) = p_{\mathcal{M}}(Y_i^{-1} Y_j \mid \mathcal{A}_{i \rightarrow j})$ directly. As the noise distribution $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \Delta_{i \rightarrow j})$ is determined by integrating Gaussian random variables along $X_i \rightarrow Y_i \rightarrow Z_{i \rightarrow j} \rightarrow Y_j \rightarrow X_j$ it can nicely be approximated by a Gaussian [21].

The problem of learning the probability distribution now comes down to learning the function $f_{\mathcal{M}} : \mathbb{R}^{|\mathcal{A}_{i \rightarrow j}|} \rightarrow \mathbb{R}^6$, $\mathcal{A}_{i \rightarrow j} \mapsto Z_{i \rightarrow j}$, from the training data. A flexible model for solving such non-linear regression problems given noisy observations is the popular Gaussian process (GP) approach. The main feature of the Gaussian process framework is, that the observed data points are explicitly included the model and, thus, no parametric form of $f_{\mathcal{M}}$ needs to be specified. Data points can be added to the training set at any time, which facilitates incremental and online learning. Due to space constraints, we refer the interested reader to work by Rasmussen [22] for technical details about GP regression. For simplicity, we assume independence between all 12 free components of $f_{\mathcal{M}}(\mathcal{A}_{i \rightarrow j})$ and consider the functional mapping for each component separately. Due to this simplification, we cannot guarantee that the prediction corresponds to a valid, homogeneous transformation matrix. In practice, however, invalid transformations occur only rarely and they lie close to similar, valid transformations, such that a simple normalization step resolves the problem.

C. Learning a Factorized Full Body Model

We seek to find the best factorized model according to Eqn. 3 and, thus, require a suitable optimization criterion. Given a training set \mathcal{D} of independent, time-indexed actions and their corresponding observations, $\mathcal{D} = \{(Y_i^t, Y_j^t, \mathcal{A}_{i \rightarrow j}^t)\}_{t=1}^T$, or, equivalently for our purposes, $\{(Z_{i \rightarrow j}^t, \mathcal{A}_{i \rightarrow j}^t)\}_{t=1}^T$, the data likelihood $p(\mathcal{D} \mid \mathcal{M})$ under a local model \mathcal{M} can directly be computed from its probability distribution $p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j})$ as

$$p(\mathcal{D} \mid \mathcal{M}) = \prod_{k=1}^t p_{\mathcal{M}}(Z_{i \rightarrow j}^k \mid \mathcal{A}_{i \rightarrow j}^k). \quad (3)$$

In practice, this product is highly sensitive to outliers, and makes the comparison of different classes of models difficult. We therefore developed an alternative model quality measure $q(\mathcal{D} \mid \mathcal{M})$ that is proportional to both the prediction accuracy and a penalty term for model complexity:

$$\log q(\mathcal{D} \mid \mathcal{M}) := \log(1/\epsilon_{\text{pred}}(\mathcal{D} \mid \mathcal{M})) + C(\mathcal{M}) \log \theta \quad (4)$$

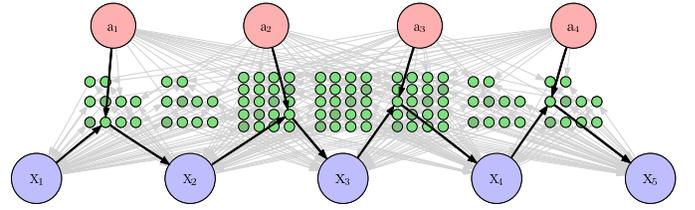


Fig. 4. In an early learning phase, the robot knows only little about its body structure, i.e., all possible local models need to be considered in parallel. From the subset of valid local models, a minimal spanning tree can be constructed which, in turn, forms a Bayesian network. This can subsequently be used as a body scheme for prediction and control.

where $C(\mathcal{M}) \in \mathbb{Z}$ is the complexity of model \mathcal{M} and $\epsilon_{\text{pred}}(\mathcal{D} \mid \mathcal{M})$ is the prediction error defined as

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

with

$$\epsilon_{\text{pred}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}, \mathcal{M}) := \int_{\mathcal{Z}} \|Z_{i \rightarrow j} - Z\| \cdot p_{\mathcal{M}}(Z' \mid \mathcal{A}_{i \rightarrow j}) dZ. \quad (6)$$

We define a local model \mathcal{M} to be $\text{valid}_{\mathcal{M}}(\mathcal{D})$ given a set of observations, if and only if its observed prediction error is below some threshold θ , i.e., $\epsilon_{\text{pred}}(\mathcal{D}) < \theta$. Our experiments revealed that a good value for θ is 3σ , where σ is the standard deviation of the sensor model.

1) *Bootstrapping*: If no prior knowledge of the robot's body scheme exists, we initialize a fully connected network model (see Figure 4), resulting in a total set of $\sum_{k=0}^m \binom{n}{2} \binom{m}{k}$ local models. Given a set of self observations, the robot can determine the validity of the local models by evaluating Eq. 7. Certain ambiguities will, however, remain even after infinitely many training samples: if, for example, $p_{\mathcal{M}_1}(Z_{1 \rightarrow 2} \mid a_1)$ has been determined to be a valid local model, then $p_{\mathcal{M}_2}(Z_{1 \rightarrow 2} \mid a_1, a_2)$ will also be. Although \mathcal{M}_1 and \mathcal{M}_2 might not be distinguishable regarding prediction accuracy, these models differ significantly in terms of complexity and therefore in model quality $q(\mathcal{D} \mid \mathcal{M})$.

2) *Finding the Network Topology*: From the superset of all valid local models $\mathbb{M}_{\text{valid}} = \{\mathcal{M}_1, \dots\}$, we seek to select the minimal subset $\mathbb{M} \subset \mathbb{M}_{\text{valid}}$ that covers all body part variables and simultaneously maximizes the overall model fit $q(\mathcal{D} \mid \mathbb{M}) := \prod_{\mathcal{M} \in \mathbb{M}} q(\mathcal{D} \mid \mathcal{M})$. It turns out that \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. Such a body spanning tree needs to cover all body parts X_1, \dots, X_n but not necessarily all action components of a_1, \dots, a_m . Note that, in order to connect all n body poses in the Bayesian network, exactly $(n - 1)$ local models need to be selected. This yields the astronomical number of $\#_{\text{structures}} = \binom{\#_{\text{local models}}}{n-1}$ possible network structures to be considered. In practice, however, simple search heuristics allow us to strongly focus the search on the

relevant parts of the structure space. 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 $\epsilon_{pred}(\mathcal{D} \mid \mathcal{M}_{1|2}) < \theta$, 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.,

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

Thus, it is sufficient to evaluate only the first k complexity layers of local models in \mathbb{M}_{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.

D. Prediction and Control

The *kinematic forward model* is directly available by noting

$$\begin{aligned} p(Y_1, \dots, Y_n \mid a_1, \dots, a_m) \\ &= \prod_i p(Y_i \mid \text{parents}(Y_i)) p(\text{parents}(Y_i) \mid a_1, \dots, a_m) \\ &= p(Y_{root}) \prod_{\mathcal{M} \in \mathbb{M}} p_{\mathcal{M}}(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}) , \end{aligned} \quad (8)$$

where Y_{root} is the position of the robot's trunk, which is serving as the coordinate origin of all other body parts. In practice, instead of a probability distribution $p(Y_1, \dots, Y_n \mid a_1, \dots, a_m)$, we rather require the maximum likelihood (ML) estimate of the resulting body posture given an action signal. This can be computed efficiently by concatenating the geometric transformations of the individual mapping functions $f_{\mathcal{M}_i}$.

Although the *inverse kinematic model* can in principle be derived by applying the rules of Bayes,

$$\begin{aligned} p(X_1, \dots, X_n \mid a_1, \dots, a_m) \\ &= \frac{p(X_1, \dots, X_n)}{p(a_1, \dots, a_m)} p(a_1, \dots, a_m \mid X_1, \dots, X_n) \\ &\propto p(a_1, \dots, a_m \mid X_1, \dots, X_n), \end{aligned} \quad (9)$$

it is in general difficult to determine the maximum likelihood (ML) estimate for the action signal a_1, \dots, a_m that is supposed to generate a given target body posture X_1, \dots, X_n . 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 $\nabla f(\mathbf{a})$ of the forward model as

$$\nabla f(\mathbf{a}) = \left[\frac{\partial f(\mathbf{a})}{\partial a_1}, \dots, \frac{\partial f(\mathbf{a})}{\partial a_m} \right]^T . \quad (10)$$

A gradient descent algorithm can then be used to minimize $f(\mathbf{a})$ and thereby to iteratively control the manipulator to its target position [7].

E. Failure Awareness and Life-Long Model Adaptation

Until now, we have assumed that the robot's physiology remains 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 insight requires that the robot revises

parts of its experience over time, allowing it to discriminate between earlier and more recent observations. This enables the robot to detect changes in its physiology 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 schemes at different time scales. Consider, for example, a robot that uses an accurate pre-programmed model over a long period of time, but simultaneously is able to create and use a short-term model that takes over as soon as the body structure of the robot changes occur (which could be as little as the displacement of one visual marker). From a formal point of view, time is simply another dimension in the model space which can be included in the definition of local models.

A *temporal local model* \mathcal{M}^T describes the geometric relationship $p_{\mathcal{M}}^T(Z_{i \rightarrow j} \mid \mathcal{A}_{i \rightarrow j}, T)$ between two observed body parts Y_i and 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 in the bootstrapping case now grows exponentially in time yielding the immense upper bound of $\sum_{k=0}^m \binom{n}{k} \binom{m}{k} 2^{|T|}$ 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 real-time application can be devised:

- 1) Changes in body physiology can be assumed to be relatively rare events.
- 2) Changes in physiology most probably happen incrementally.
- 3) Whatever local models were useful in the past, it is likely that similar (or maybe even the same) local models will be useful in the future.

Because of the first assumption it is not necessary to consider new local models as long as the current body scheme still yields a high prediction accuracy. Only when one of the local models of the current body scheme becomes invalid, incremental learning (assumption 2) has to be triggered. Then, according to assumption 3, it is reasonable to begin the search for new models that are similar to previously useful models. To incorporate these assumptions in the quality measure for local models, we first define the concept of relative complexity of a local model \mathcal{M}_2 given a previously used model \mathcal{M}_1 as

$$C(\mathcal{M}_2 \mid \mathcal{M}_1) := d(\mathcal{M}_2, \mathcal{M}_1),$$

where $d(\cdot, \cdot)$ is a (data-independent) similarity metric between two local models and $C(\mathcal{M}_2 \mid \mathcal{M}_1) \in \mathbb{Z}$. In practice, $d(\cdot, \cdot)$ can for example be defined as the ratio of shared nodes between two local models in the Bayesian network. The refined version of the model quality measure $q_2(\mathcal{D} \mid \mathcal{M}_1, \mathcal{M}_2)$ of some recent observations \mathcal{D} given a newly sampled model \mathcal{M}_2 as a replacement for an invalidated previous model \mathcal{M}_1 can then be defined as

$$\begin{aligned} \log q_2(\mathcal{D} \mid \mathcal{M}_1, \mathcal{M}_2) &:= \log(1/\text{error}_{prediction}(\mathcal{D})) \\ &+ C(\mathcal{M}_2 \mid \mathcal{M}_1) \log \theta \\ &+ \log |T_{\mathcal{M}_2}| . \end{aligned} \quad (11)$$

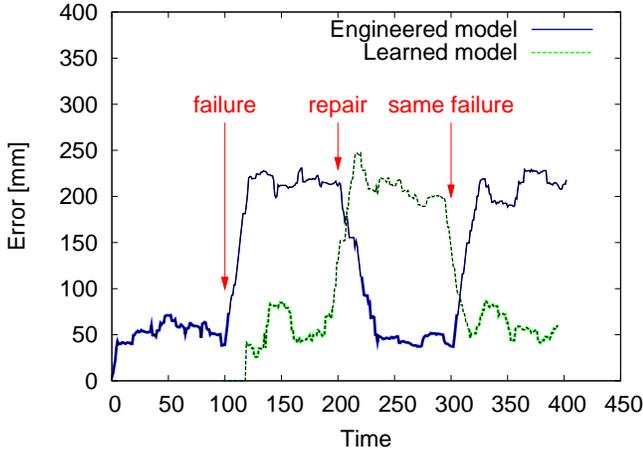


Fig. 5. At $t = 100$, a joint gets blocked, which causes the initial local model $p_{\text{engineered}}(Z_{6 \rightarrow 7} | a_4)$ to produce substantially larger prediction errors. At $t = 126$, the robot samples a new local model $p_{\text{learned}}(\Delta 6 \rightarrow 7)$ as replacement.

Please note that, by construction, the quality measure of two local models with different relative complexity have no overlapping ranges in model quality independently of the observation data \mathcal{D} , i.e.,

$$C(\mathcal{M}_1 | \mathcal{M}_3) < C(\mathcal{M}_2 | \mathcal{M}_3) \\ \iff \forall \mathcal{D} : q(\mathcal{D} | \mathcal{M}_1) > q_2(\mathcal{D} | \mathcal{M}_2) . \quad (12)$$

It is, like in the static case, sufficient to sample and evaluate only the first k complexity layers of local models until a minimum spanning tree is found. By definition of the quality function, this minimum spanning tree is then by construction the global maximum of overall model quality.

IV. EXPERIMENTS

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) physiological changes are detected confidently (blocked joints / deformations),
- 2) the body scheme is updated automatically without human intervention, and
- 3) the resulting body scheme can be used for accurate prediction and control.

The robot used to carry out the experiments is equipped with a 6-DOF manipulator composed of Schunk PowerCube modules. The total length of the manipulator is around 1.20m. With nominal noise values of ($\sigma_{\text{joints}} = 0.02^\circ$), the reported joint positions of the encoders were considered to be sufficiently accurate to compute the ground truth positions of the body parts from the known geometrical properties of the robot. Visual perception was obtained by using a Sony DFW-SX900 FireWire-camera at a resolution of 1280x960 pixels. On top of the robot's joints, 7 black-and-white markers were attached (see Figure 1), that were detectable by the ARToolkit vision

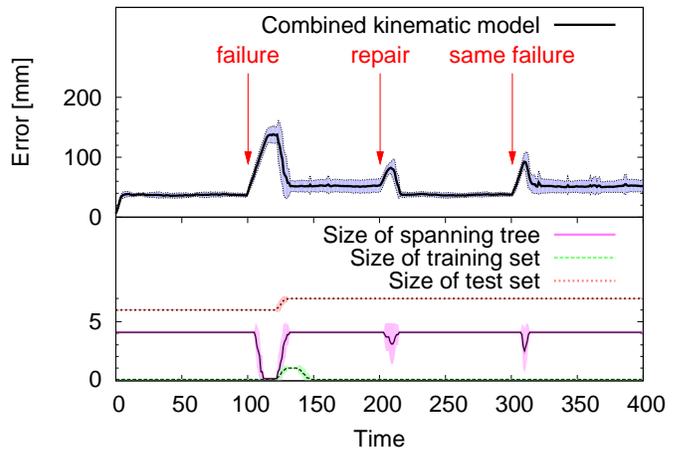


Fig. 6. The absolute prediction error of the combined kinematic model $p(Z_{1 \rightarrow 7} | a_1, \dots, a_4)$ of our 6-DOF manipulator. This model is composed of 6 individual local models of which one is replaced by a newly learned model at $t = 126$ (cmp. Figure 5). As can be seen from the plot, the prediction accuracy recovers quickly after each of the three external events.

module [20]. Per image, the system perceives the unfiltered 6D poses of all detected markers. The standard deviation of the camera noise was measured to $\sigma_{\text{markers}} = 44\text{mm}$ in 3D space, which is acceptable considering that the camera was located two meters apart from robot.

We averaged the prediction error over a test set of the latest $|\mathcal{D}_{\text{testing}}| = 15$ data samples. New local models were trained with $|\mathcal{D}_{\text{training}}| = 30$ succeeding training samples after the model was instantiated. In order for a local model to be valid, its translational and rotational error on the test set needed to be below a threshold of $\theta_{\text{trans}} = 3\sigma_{\text{trans}} = 150\text{mm}$ and $\theta_{\text{rot}} = 3\sigma_{\text{rot}} = 45^\circ$, with σ_{trans} and σ_{rot} as the standard deviation of the translational and rotational observation noise, respectively. New local models were only sampled when no valid spanning tree could be constructed for $|\mathcal{D}_{\text{testing}}|$ succeeding time steps, as this is the time it takes to replace most if not all (because of possibly missing observations) data samples of the test set. Note that otherwise it could happen that available local models cannot be selected because the test set temporarily consists of data samples partly observed just before and partly after a change in physiology.

A. 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 a 2-DOF robot of which the kinematic model is perfectly known.

Figure 7 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 scheme converges robustly within the first 10 observations. After about

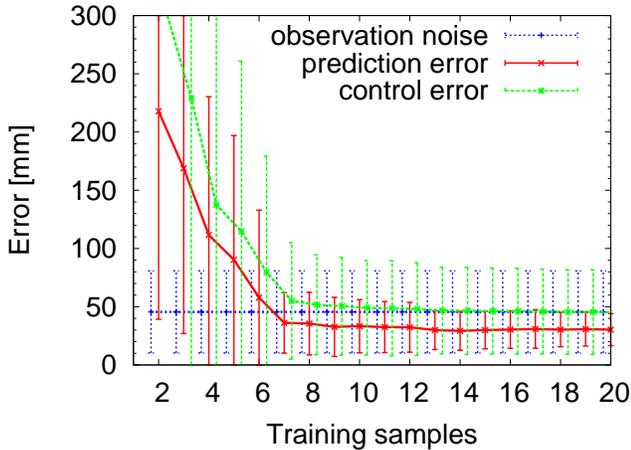


Fig. 7. Prediction and control errors for a kinematic model that is learned from scratch. Already after 7 samples, the average prediction error is lower than the average localization error of the visual markers.

15 training samples, the accuracy of the predicted body part positions even outperformed the accuracy of the direct observations. The latter is a remarkable result as it means that, although all local models are learned from noisy observations, the resulting model is able to blindly predict positions that are more accurate than 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 robot’s end effector. We learned the full body scheme 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. 50mm, see Figure 7), i.e. slightly higher than the prediction error alone.

B. Scenario 1: Joint stuck

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 (X_6 and X_7) would potentially be visible on the camera image. This sequence was sent to the robot and after each motion command, the observed marker positions $\langle Y_1, \dots, Y_n \rangle$ were recorded. In the rare case of a anticipated or a real (self-)collision during execution, the robot stopped and the sample was rejected. Careful 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 robot’s end-effector joint a_4 , such that it could not move, and again recorded a log-file. Note that we allow arbitrary 3D motion (just constrained by the geometry of the manipulator) and thus do not assume full visibility of the markers.

An automated test procedure was then used to evaluate the performance and robustness of our approach. For each of the 20 runs, a new data set was sampled from the recorded 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 5 shows the prediction error 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 results for 20 reruns of this experiment are given in Figure 6. The hand-tuned initial geometrical model evaluates to an averaged error at the end-effector of approx. 37mm. 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 51mm. 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 much fewer data samples than learning a new model (see Table I). At $t = 300$, the same quick adaption can be observed when the joint gets blocked again.

C. Scenario 2: Deformed limb

In a second experiment¹, 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 robot’s gripper at different locations (see Figure 1).

¹A demonstration video of this experiment can be found on the internet at <http://www.informatik.uni-freiburg.de/~sturm/media/resources/public/zora-7dof-demo.avi>

TABLE I
EVALUATION OF THE RECOVERY TIME REQUIRED AFTER BEING EXPOSED TO DIFFERENT TYPES OF FAILURES. IN EACH OF THE 4×20 RUNS, FULL RECOVERY WAS AFTER EACH EVENT ROBUSTLY ACHIEVED.

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

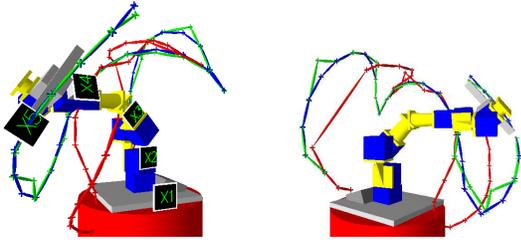


Fig. 8. The manipulator robot with a deformed limb has to follow the blue target trajectory. With a static body model, it suffers from strong derivation (red trajectory). By using our approach, the body scheme is dynamically adapted, and the trajectory is very well approached (green trajectory).

Although the overall result closely resembles the case of a blocked joint, there are a few interesting differences. After the tool gets displaced at $t = 100$, on average two local models need to be sampled because the first one is not sufficient.

Also note that it takes much more training samples for the GPs to learn and validate the underlying probability distribution $p(Z_{6 \rightarrow 7} | a_4)$ (see Table I). The prediction accuracy of the whole system closely resembles the levels as in the case of the blocked joint: On average, we measured after recovery an accuracy of 47mm.

D. Controlling a Deformed Robot

Finally, we ran a series of experiments to verify that dynamically maintained body schemes can be used for accurate positioning and control. The experiments were executed on a 4-DOF manipulator in simulation.

We defined a 3D trajectory consisting of 30 way-points that the manipulator should approach by inverse kinematics using its current body scheme, see Figure 8. 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° , the manipulator with a static body scheme was significantly off course, leading to an average positioning accuracy of 189.35mm. With dynamic adaptation enabled, the precision settled at 15.24mm. This shows that dynamic model adaption enables a robot to maintain a high positioning accuracy after substantial changes to its body physiology.

V. CONCLUSION

In this paper, we presented a novel approach to life-long body scheme adaptation for a robotic manipulation system. Our central idea is to continuously learn a large set of local kinematic models using non-parametric regression and to search for the best arrangement of these models to represent the full system.

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. Challenging topics for further investigation include developing an active exploration strategy, learning from marker-less observations, point-like features, or

range observations and learning for fully unobservable parts of the robot.

ACKNOWLEDGMENTS

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

REFERENCES

- [1] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, June 2005.
- [3] E. M. Rosales and Q. Gan, "Forward and inverse kinematics models for a 5-dof pioneer 2 robot arm," University of Essex, UK, Tech. Rep. CSM-413, 2004.
- [4] J. Kolter and A. Ng, "Learning omnidirectional path following using dimensionality reduction," in *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, June 2007.
- [5] A. Dearden and Y. Demiris, "Learning forward models for robots," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, 2005, pp. 1440–1445.
- [6] D. Grimes, R. Chalodhorn, and R. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Proc. of the Robotics: Science and Systems (RSS)*, Philadelphia, USA, 2006.
- [7] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Pasadena, CA, USA, 2008, to appear.
- [8] Y. Yoshikawa, K. Hosoda, and M. Asada, "Binding tactile and visual sensations via unique association by cross-anchoring between double-touching and self-occlusion," in *Proc. of the International Workshop on Epigenetic Robotics*, Genoa, Italy, 2004, pp. 135–138.
- [9] Y. Yoshikawa, Y. Tsuji, K. Hosoda, and M. Asada, "Is it my body? – body extraction from uninterpreted sensory data based on the invariance of multiple sensory attributes –," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Sendai, Japan, 2004.
- [10] P. Morasso and V. Sanguineti, "Self-organizing body-schema for motor planning," *Journal of Motor Behavior*, vol. 26, pp. 131–148, 1995.
- [11] L. Natale, "Linking action to perception in a humanoid robot: A developmental approach to grasping," Ph.D. dissertation, University of Genoa, Italy, May 2004.
- [12] J. Ting, M. Mistry, J. Peters, S. Schaal, and J. Nakanishi, "A bayesian approach to nonlinear parameter identification for rigid body dynamics," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.
- [13] N. Roy and S. Thrun, "Online self-calibration for mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1999.
- [14] J. C. Bongard, V. Zykov, and H. Lipson, "Automated synthesis of body schema using multiple sensor modalities," in *Proc. of the Int. Conf. on the Simulation and Synthesis of Living Systems (ALIFEX)*, 2006.
- [15] C. Nabeshima, M. Lungarella, and Y. Kuniyoshi, "Timing-based model of body schema adaptation and its role in perception and tool use: A robot case study," in *Proc. of the IEEE International Conference on Development and Learning (ICDL2005)*, 2005, pp. 7–12.
- [16] B. J., Z. V., and L. H., "Resilient machines through continuous self-modeling," *Science*, vol. 314., no. 5802, pp. 1118–1121, 2006.
- [17] B. J. and L. H., "Automated reverse engineering of nonlinear dynamical systems," in *Proc. of the Nat. Academy of Science*, vol. 104, no. 24, 2007, p. 9943.
- [18] C. Stachniss, D. Hähnel, W. Burgard, and G. Grisetti, "On actively closing loops in grid-based FastSLAM," *Advanced Robotics*, vol. 19, no. 10, pp. 1059–1080, 2005.
- [19] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, January 2000.
- [20] M. Fiala, "Artag, a fiducial marker system using digital techniques," National Research Council Canada, Tech. Rep., 2004.
- [21] A. T. Ihler, E. B. Sudderth, W. T. Freeman, and A. S. Willsky, "Efficient multiscale sampling from products of gaussian mixtures," in *NIPS*, 2003.
- [22] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2006.