Context Estimation and Learning Control through Latent Variable Extraction: From discrete to continuous contexts

Georgios Petkos and Sethu Vijayakumar

Abstract— **Recent advances in machine learning and adaptive motor control have enabled efficient techniques for online learning of *stationary* plant dynamics and it's use for robust predictive control. However, in realistic domains, system dynamics often change based on unobserved external contexts such as work load or contact conditions with other objects. Previous multiple model approaches to solving this problem are restricted to finite, discrete contexts without any generalization and have been tested only on linear systems. We present a framework for estimation of context through hidden latent variable extraction – solely from experienced (non-linear) dynamics. This work refines the multiple model formalism to bootstrap context separation from context-unlabeled data and enables *simultaneous* online context estimation, dynamics learning and control based on a consistent probabilistic formulation. Most importantly, it extends the framework to a *continuous* latent model representation of context under specific assumptions of load distribution.**

I. INTRODUCTION

The dynamics of a system often depend on an unobserved external context. An example of unobserved external context that results in non-stationary dynamics is the work load of a robot manipulator. The resultant dynamics of the robot arm change as it manipulates objects with different physical properties, e.g. mass, shape or mass distribution. The study of adaptive control [8] has provided a multitude of methods that could be used in cases of non-stationary dynamics. However, if the dynamics switch back and forth, e.g. if manipulating a set of tools for executing various tasks, classic adaptive control methods are inadequate since they result in large errors and instability during the period of adaptation; moreover, readapting every time is a suboptimal and inefficient strategy that unlearns the dynamics of the previous contexts. A proposed solution is the use of multiple models, each of which is appropriate for a different context. However, existing work on the multiple models paradigm [3], [10], [5], [7] does not cope well with issues concerning the choice of correct number of models, detection of novel contexts and use of knowledge from previously learned models to new contexts. Furthermore, the actual number of discrete contexts may grow indefinitely with time as new situations appear. Most prior work on estimating contexts from movement data rely heavily on analytical rigid body dynamics and estimation of a few, heavily constrained parameters of the full body dynamics[4]. This approach fails when deriving analytical dynamics is complicated or not feasible.

Our approach will formulate a probabilistic model that represents the context as a latent (switching or continuous) variable. This framework allows us to estimate the context online based only on the learned inverse dynamics models using Markovian filtering. Further, an Expectation-Maximization procedure is used to bootstrap the distinction of contexts from context-unlabeled data. In Section II, we briefly discuss single model learning and control under a single context using LWPR, an efficient online algorithm. We then talk about the multiple model paradigm and discuss context estimation, control and data separation under multiple discrete contexts in Section III. We then show in Section IV, using knowledge about analytical dynamics, that it is possible to reformulate the discrete context scenario to a continuous latent model representation where the generalizations to new contexts (outwith the already learned models) holds under specific assumptions of the load distribution. To the best of our knowledge, this is the first work that deals with learning control under *continuously* varying contexts.

II. LEARNING DYNAMICS FOR CONTROL

Anthropomorphic robotic systems have complex kinematic and dynamic structure, significant non-linearities and hard to model non-rigid body dynamics; hence, deriving reliable analytical models of their dynamics can be cumbersome and/or inaccurate. We take the approach of learning dynamics for control from movement data (see Fig. 1 for a graphical model representation); typically the *inverse* dynamics model for predicting desired torques. The inverse model maps current states Θ_t and the next desired states Θ_{t+1} to the command τ_t that results in the transition between these states:

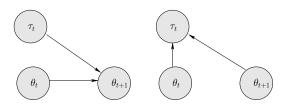$$\tau_t = g(\Theta_t, \Theta_{t+1}) . \tag{1}$$

Fig. 1.   The forward and inverse model

The inverse model shown in Fig. 1(right) can be used in many control settings; the most common being to use it as

part of a composite controller. Given a desired trajectory, $\Theta^*_{1:T}$, the composite control law computes the command as

$$\tau_t = g(\Theta^*_t, \Theta^*_{t+1}) + A\,(\Theta^*_t - \Theta_t)\,, \qquad (2)$$

where A is the gain matrix. This is a combination of a feedforward command that uses the inverse model and a feedback command that takes into account the actual state of the system. The more accurate the inverse model is, the lower the feedback component of the command will be, i.e., the magnitude of the feedback command can be used as a measure of the accuracy of the inverse model. Furthermore, good predictive models allow us to use low feedback gains, resulting in a highly compliant system without sacrificing the speed and accuracy of the movements.
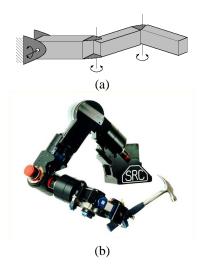

(a)

(b)

Fig. 2.   (a) Simulated 3DOF arm (b) 7DOF SARCOS dextrous arm

Typically, in robotic systems with proprioceptive and torque sensing, at each time step t we "observe" a state transition and an applied torque signal summarized in the triplet $(\Theta_t, \Theta_{t+1}, \tau_t)$, i.e., we have access to the true applied control command (which was generated via composite control). To learn the inverse dynamics, we need a *non-linear, online* regression technique which also provides error bounds that we may use for context identification. We use the Locally Weighted Projection Regression (LWPR) [9] – an algorithm which is extremely robust and efficient for incremental learning of non-linear models in high dimensions. An LWPR model uses a set of linear models, each of which is accompanied by a locality kernel (usually a gaussian) that defines the area of validity of the linear model. For an input $\boldsymbol{x}$, if the output of the kth local model is written as $y_k(\boldsymbol{x})$ and the locality kernel activation is $w_k(\boldsymbol{x})$, the combined prediction of the LWPR model, $\hat{y}$, is

$$\hat{y}(\boldsymbol{x}) = \frac{1}{W}\sum_k w_k(\boldsymbol{x})\,y_k(\boldsymbol{x})\,, \quad W = \sum_k w_k(\boldsymbol{x})\,. \quad (3)$$

The parameters of the local linear models and locality kernels are adapted online and also local models are added on an as needed basis. Furthermore, LWPR provides statistically sound input dependent confidence bounds on its predictions
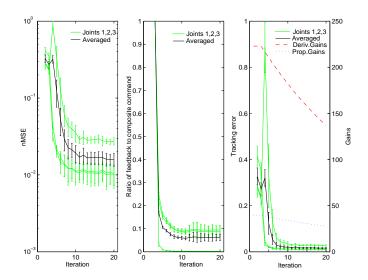

Fig. 3.   Results on learning single context dynamics. Left: test error. Middle: contribution of error-correcting feedback command. Right: Tracking error

and employs Partial Least Squares (PLS) to deal with high dimensional inputs. For more details about LWPR, see [9].

A. Experiments in Learning Dynamics for Single Context

We verify the ability to learn the inverse model online with LWPR and show that the model can successfully be used for control. We demonstrated this for a simulated 3 DOF robot arm [1] (see Fig. 2(a)) as well as on the 7 DOF anthropomorphic SARCOS robot arm (Fig. 2(b)). The statistics are accumulated and shown briefly for the simulated arm, for more details please see [2]. The task of the arm was to follow a simple trajectory planned in joint angle space, consisting of a superposition of sinusoids with different phase shifts. 20 iterations of the trajectory were repeated: during the first four iterations, pure feedback (PD) control was used to control the arm, while at the next 16 iterations, a composite controller using the inverse model being learned was used. The gains were lowered as training proceeded. The procedure was executed six times and repeated for six different contexts for accumulating the statistics. Fig. 3(left) plots the normalized mean squared error between the torques predicted by the LWPR model and the true torques experienced on the test data (i.e., the data that was held out from the training), which shows a quick drop as training proceeds and settles at a very low value averaged over all trials. The contribution of the error-correcting feedback command to the feedforward command (see Fig. 3(middle)) is low, vouching for the accuracy of the learnt model while being used for control. Furthermore, the tracking error (Fig. 3(right)) is very low and improves significantly when we switch to composite control. For the detailed statistics on the online dynamics learning of the 7 DOF SARCOS robot arm and tracking results on a pattern eight task, readers are referred to [9].

[1]Simulations performed using ODE and OpenGL

III. THE MULTIPLE MODEL PARADIGM

Although we have verified the ability to learn dynamic models and perform control under a single context, the main emphasis of this work is the ability to cope with varying contexts. The multiple model paradigm copes with the issue of non-stationary dynamics by using a set of models, each of which is specialized to a different context. A schematic of a generic multiple model paradigm is shown in Fig. 4. The
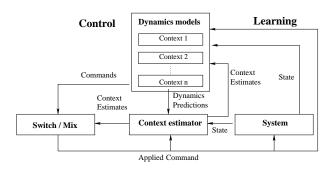

Fig. 4.   Schematic of a multiple model paradigm

observed dynamics of the system are compared to the prediction of each learned model to identify the current context. The context estimates are used for selecting the model to use for control and for training. All existing multiple model paradigms roughly follow the same plot. Some of the existing models are Modular Selection and Identification for Control (MOSAIC) [3], Multiple Paired Forward and Inverse Models [10] and Multiple Model Switching and Tuning (MMST) [5], [7]. The main issues that have to be tackled for using multiple discrete models for control are:

1) Infer the current context for selecting the appropriate model to use for *control*.
2) Infer the current context for selecting the appropriate model to *train* with the experienced data.
3) Figure out the appropriate *number of models* (possibly using a novelty detection mechanism).

Hence, it is clear that context estimation is of critical importance in the multiple model scenario.

A. Context Estimation

It is appropriate to formulate context estimation in a probabilistic setting to account for inaccuracies of the learnt models as well as handle transitions. The graphical model in Fig. 5(a) represents a set of inverse models corresponding to a specific number of contexts. The hidden contextual variable c_t is discrete and indexes the different models. The inverse model in this formulation can be written as:

$$P(\tau \,|\, \Theta_{t+1}, \Theta_t, c_t = i) = \mathcal{N}(\tau^{(i)}(\Theta_{t+1}, \Theta_t),\, \sigma^{(i)}(\Theta_{t+1}, \Theta_t)),$$

where $\tau^{(i)}$ is the command predicted by the LWPR model corresponding to the ith context and $\sigma^{(i)}$ is some estimate of the variance, which can be either set to a predetermined constant or based upon the input dependent confidence bounds provided by LWPR. Also, if there is no knowledge about the prior probability of contexts, we can assume that different
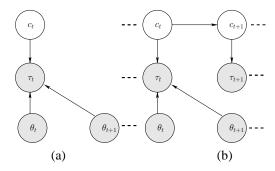

Fig. 5.   Multiple models and hidden contexts

contexts have equal prior probabilities $p(c_t)$. Under this probabilistic formulation, context estimation is just inferring the posterior of c_t given a state transition and the command that resulted in this transition:

$$P(c_t = i \,|\, \Theta_t, \Theta_{t+1}, \tau_t) \propto P(\tau_t \,|\, c_t = i, \Theta_t, \Theta_{t+1})P(c_t = i). \quad (4)$$

Context estimates are very sensitive to the accuracy of the inverse models. They can be improved by acknowledging that contexts do not change too frequently. We can introduce a temporal dependency between contexts $p(c_{t+1} \,|\, c_t)$ with an appropriate transition probability between contexts that reflects our prior belief on the switching frequency to achieve much more robust context estimation. The graphical model can be reformulated as the Dynamic Bayesian Network shown in Fig. 5(b) to achieve this. Application of standard Hidden Markov Model (HMM) techniques is straightforward by using (4) as the observation likelihood in the HMM, given the hidden state $c_t = i$. A low transition probability penalizes too frequent transitions and using smoothing or Viterbi alignment produces more stable context estimates. In the experiments, we will assume a fixed transition matrix $P(c_t = j \,|\, c_t = i)$ with high value .999 for $i = j$ and .001 otherwise and use the HMM model only for filtering or smoothing, depending on whether we investigate an online or batch estimation scenario, respectively.

B. Data Separation

Context estimates are used for guiding online control and for further training of the models. However, to get these context estimates we need a mechanism for getting relatively accurate (initial) models to bootstrap the context estimation procedure. Most of the existing multiple model paradigms do not give a satisfying answer to this issue. MMST assumes that relatively good models are available from the beginning, whereas MPFIM does not address this issue at all.

The problem of bootstrapping the context separation from context-unlabeled data is very similar to clustering problems using mixture of Gaussians. In fact, the context variable can be interpreted as a latent mixture indicator and each inverse model contributes a mixture component to give rise to the mixture model of the form $P(\tau_t \,|\, \Theta_t, \Theta_{t+1}) = \sum_i P(\tau_t \,|\, \Theta_t, \Theta_{t+1}, c_t = i)\ P(c_t = i)$. Clustering with mixtures of Gaussians is usually trained using Expectation-Maximization (EM), where initially the data are labeled
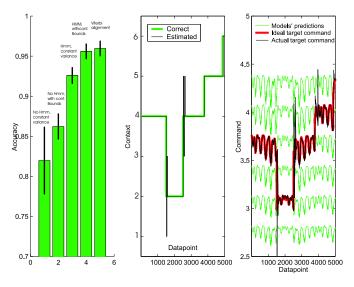
Fig. 6. Discrete context estimation under randomly switching dynamics

with random responsibilities Then every mixture component is trained on its assigned (weighted) data (M-step) and afterwards the responsibilities for each data point is recomputed by setting them proportional to the likelihoods for each mixture component (E-step). Iterating this procedure, each mixture component will specialize on different parts of the data and the responsibilities encode the learned cluster assignments.

We will apply a common variant of the EM-algorithm where responsibilities are computed greedily, i.e., where the data is hard assigned to the mixture component with maximal likelihood instead of weighted continuously with the component's likelihood in the M-step. In our case, the likelihood of a data triplet $(\Theta_t, \Theta_{t+1}, \tau_t)$ under the ith inverse model is $P(\tau_t \,|\, \Theta_t, \Theta_{t+1}, c_t = i)$, which is a Gaussian with either fixed variance or the variance given by LWPR's confidence bounds.

C. Experiments With Multiple Discrete Models

The context estimation methods described in Section III-A were used for online estimation and control with six separately learnt contexts. Random switches between the six contexts were performed in the simulation, where at every time step we switch to a random context with probability .001 and stay in the current context otherwise. The context estimates were used online for selecting the model that will provide the feed-forward commands.

We have two classes of experiments: one where we use HMM filtering of the contextual variable and the other where it is not used. We also have two choices for the variance of the observation model: one where we use a constant (found empirically) and the other, where we use the more principled confidence bounds provided by LWPR. The simulation was run for 10 iterations.

The percentage of accurate online context estimates for the four cases along with offline Viterbi alignment are shown in the Fig. 6(left). Fig. 6(middle) gives an example of how

the best context estimation method that we have, the HMM filtering using LWPR's confidence bounds, performs when used for online context estimation and control. Sometimes the context estimation lags behind a few time steps when there are context switches, which is a natural effect of online filtering (as opposed to retrospect smoothing). Fig. 6(right) presents an example of the predictions of the six models along with the actually applied command.

The performance of online context estimation and control is close to the control performance we achieved for the single context displayed in Fig. 3. Using the HMM filtering based on LWPR's confidence bounds, the average tracking error over the 10 cycles was 0.0019 and the ratio of feedback PD control was 0.074.

In another analysis, automatic separation of data to contexts was tested. We ran the simulation switching randomly between two different contexts, collected the data and executed 6 iterations of the EM-like algorithm described in Section III-B. The evolution of the assignment of datapoints to
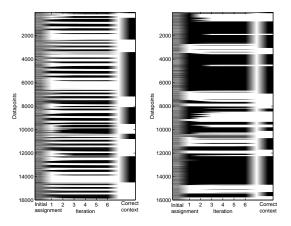

Fig. 7. Automatic separation of datapoints to contexts. Left: the non-temporal model is used for the E-step. Right: the temporal model is used in the E-step

contexts can be seen in Fig. 7. On the left, the non-temporal model has been used for context estimation, whereas on the right, the temporal model has been used. The first column displays the random initial assignment of datapoints to contexts, whereas the last column displays the correct assignment. As can be seen in the plot on the right, at the end of the last iteration, most of the datapoints are grouped correctly (84% of the data was classified correctly). The learned models were then used for online control and further online training. Twelve iterations of the trajectory were executed, with random switches between the two contexts. Accuracy in context estimation was 88% while the tracking error was 0.0051 and the ratio of feedback PD control was 0.23. The errors are slightly higher than in the case where models were trained using labeled data, but this is satisfying considering the fact that we started with unlabeled data.

IV. AUGMENTED MODEL FOR CONTINUOUS CONTEXTS

The multiple model paradigm has several limitations. First of all, the right number of discrete models needs to be known

TABLE I

LINEARITY OF THE DYNAMICS MODEL IN THE INERTIAL PARAMETERS

If $\mathcal{T}$ is the kinetic energy, $\mathcal{U}$ is the potential energy of the system and we define a Lagrangian $\mathcal{L} = \mathcal{T} - \mathcal{U}$, the dynamics of the system is given by

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \tag{5}$$

where $q_1, q_2...q_n$ is a set of generalized coordinates (here, the joint angles) and $\tau_1, \tau_2...\tau_n$ denote the so called generalized forces associated with the corresponding joint angles q_i. The generalized force τ_i is the sum of joint actuator torques, joint friction torques and other forces acting on the joint (e.g. forces induced by contact with the environment). The total kinetic energy $\mathcal{T}$ and the total potential energy $\mathcal{U}$ is just the sum of the kinetic energy and potential energies of all the links of the manipulator respectively, i.e., $\mathcal{T} = \sum_{j=1}^{n} \mathcal{T}_j$, $\mathcal{U} = \sum_{j=1}^{n} \mathcal{U}_j$ The kinetic and potential energy of the jth link is given by:

$$\mathcal{T}_j = \frac{1}{2}m_j \dot{p}_j^T \dot{p}_j + m_j l_j \dot{p}_j^T S(\omega_j) + \frac{1}{2}\omega_j^T I_j \omega_j \quad , \quad \mathcal{U}_{l_i} = -m_j g_0^T p_j - m_j g_0^T l_j \tag{6}$$

where m_j is the total **mass of link** j, p_j is the position vector of the center of mass of link j, ω_j is the rotational velocity of link j, $S(\omega_j)$ is a 3×3 skew-symmetric matrix that depends on ω_j, l_j is the position vector of the center of mass of the link from the origin of the frame of the link, g_0 is the gravity acceleration vector, I_j is the **inertia tensor of link** j measured at the origin of the reference frame of the link. Substituting (6) in the Lagrangian and with some rearrangement, we can see that the Lagrangian has a linear relationship to the set of inertial parameters:

$$\pi = [m_1, m_1 l_{1x}, m_1 l_{1y}, m_1 l_{1z}, I_{1xx}, I_{1xy}, ..., m_n, m_n l_{nx}, m_n l_{ny}, m_n l_{nz}, I_{nxx}, ..., I_{nzz}]$$

In short, the Lagrangian can be written in the form:

$$\mathcal{L} = g(q, \dot{q})\pi$$

Since the inertial parameters in π do not depend on time or $\dot{q}$ then the dynamics equation for joint i is:

$$\pi \frac{d}{dt}\frac{\partial g(q, \dot{q})}{\partial \dot{q}_i} - \pi \frac{\partial g(q, \dot{q})}{\partial q_i} = \tau_i$$

Thus, the dynamics can be written in the form

$$\tau_i = y_i(q, \dot{q}, \ddot{q})\pi \tag{7}$$

---

and estimating this is non-trivial. Realistically, novel contexts appear quite often and to cope with this, a novelty detection mechanism is needed. However, even with a very robust novelty detection mechanism, we may end up with a very large number of models, since in theory, possible contexts are infinite. Moreover, it is better if we can generalize between contexts and most multiple model paradigms do not provide an obvious way to do this.

All these issues can be circumvented if we replace the set of models with a single unique model that takes as input *continuous* hidden contextual variables, i.e., instead of a set of g_is corresponding to different contexts, a single inverse model G is used:

$$\tau_t = G(\Theta_t, \Theta_{t+1}, c_t) . \tag{8}$$

Here, c_t is not a discrete variable that indexes different models but a set of continuous variables that describe the context. The probabilistic model of the inverse dynamics would then be:

$$P(\tau \mid \Theta_t, \Theta_{t+1}, c_t) = \mathcal{N}(G(\Theta_t, \Theta_{t+1}, c_t), \ \sigma(\Theta_t, \Theta_{t+1}, c_t)) . \tag{9}$$

A possibility for learning the augmented model is to follow the same procedure as in the discrete case for learning the models, i.e., apply an EM like procedure. However, the relationship of the contextual variables to the output of the augmented model could be arbitrary, making learning in such a setting a very difficult task. It is imperative to exploit any prior knowledge about the relationship of the inverse model to appropriate contextual variables.

For the case of manipulation of objects with a robot arm, this is possible. It can be shown that the dynamics of a robot arm have a linear relationship to the inertial properties of the manipulator links. In other words, the inverse dynamics can be written in the form:

$$\tau = Y(q, \dot{q}, \ddot{q})\pi \tag{10}$$

or for a specific context r:

$$\tau_r = Y_r(q, \dot{q}, \ddot{q})\pi_r \tag{11}$$

where q, $\dot{q}$ and $\ddot{q}$ denote joint angles, velocities and accelerations respectively. This relationship can be derived based on fundamentals of robot dynamics [6], [1] as shown in Table I. This equation splits the dynamics in two terms. $Y(q, \dot{q}, \ddot{q})$ is a term that depends on kinematics properties of the arm such as link lengths, direction of axis of rotation of joints and so on. This is a very complicated and highly non-linear function of joint angles, velocities and accelerations. The term π is a high dimensional vector containing all inertial parameters of all links of the arm (see Table I).

Now, lets consider that we model the dynamics of the arm as the manipulated object being the last link of the arm. Then, manipulating different objects is equivalent to changing the physical properties of the last link of the arm. Under the assumption that $Y_r(q, \dot{q}, \ddot{q})$ is constant between different models, we could use a set of learned models, with known inertial parameters π_r to infer an augmented model that predicts the dynamics for any possible context π. Since $Y_r(q, \dot{q}, \ddot{q})$ is constant between contexts, then the augmented

In our probabilistic setting, the augmented inverse model is

$$\tau_t = G(\Theta_t, \Theta_{t+1}, c_t) = A(\Theta_t, \Theta_{t+1}) + B(\Theta_t, \Theta_{t+1})c_t + \eta \tag{12}$$

where $A(\Theta_t, \Theta_{t+1})$ and $B(\Theta_t, \Theta_{t+1})$ are estimated from the models used for forming the augmented model and $\eta = \mathcal{N}(0, \Sigma_{obs})$. Σ_{obs} is estimated from the confidence bounds of the inverse models that form the augmented model. Also, the transition model for the context needs to be defined. Since we believe that the context does not change too often, this is set to:

$$c_{t+1} = c_t + \zeta \tag{13}$$

where $\zeta = \mathcal{N}(0, \Sigma_{tr})$ with Σ_{tr} set to a very small value.
Based on the defined model, we can write down the inference for the temporal Bayesian network using the augmented inverse model. For control, only filtered estimates (a la Kalman filtering) can be used.
We want to compute $p(c_t \,|\, \tau_{1:t+1}, \Theta_{1:t+1})$ using the estimate at the previous time step $p(c_{t-1} \,|\, \tau_{1:t}, \Theta_{1:t})$ and the new evidence τ_{t+1} and Θ_{t+1}. The previous estimate $p(c_{t-1} \,|\, \tau_{1:t}, \Theta_{1:t})$ is defined as:

$$p(c_{t-1} \,|\, \tau_{1:t}, \Theta_{1:t}) = \mathcal{N}(\mu_{t-1\,|\,t}, \Sigma_{t-1\,|\,t}) \tag{14}$$

Estimates for the next time step $p(c_t \,|\, \tau_{1:t+1}, \Theta_{1:t+1})$ are obtained in a recursive way in two steps. The first is the **prediction** step where, $p(c_t \,|\, \tau_{1:t}, \Theta_{1:t})$ is computed using the filtered estimate on the previous time step and the transition model $p(c_{t+1} \,|\, c_t)$, without taking into account evidence at time $t+1$:

$$p(c_t \,|\, \tau_{1:t}, \Theta_{1:t}) = \mathcal{N}(\mu_{t\,|\,t}, \Sigma_{t\,|\,t}) \tag{15}$$

where $\mu_{t\,|\,t} = \mu_{t-1\,|\,t}$ and $\Sigma t \,|\, t = \Sigma_{t\,|\,t} + \Sigma_{tr}$. Then, the **filtered estimate** modifies the predicted estimates using the observation at the time $t+1$ as (dependency of A and B on the state transition is omitted for compactness):

$$p(c_t \,|\, \tau_{1:t+1}, \Theta_{1:t+1}) = \mathcal{N}(\mu_{t\,|\,t+1}, \Sigma_{t\,|\,t+1}) \tag{16}$$

where,

$$\mu_{t\,|\,t+1} = \mu_{t\,|\,t} + \Sigma_{t\,|\,t}B^T(B\Sigma_{t\,|\,t}B^T + \Sigma_{obs})^{-1}(\tau_{t+1} - A - B\mu_{t\,|\,t}) \tag{17}$$

$$\Sigma_{t\,|\,t+1} = \Sigma_{t\,|\,t} - \Sigma_{t\,|\,t}B^T(B\Sigma_{t\,|\,t}B^T + \Sigma_{obs})^{-1}B\Sigma_{t\,|\,t} \tag{18}$$

model $G(\Theta_t, \Theta_{t+1}, c_t)$ is simply:

$$G(\Theta_t, \Theta_{t+1}, c_t) = Y(q, \dot{q}, \ddot{q})\pi_r = \tau \tag{19}$$

where state transitions have been appropriately replaced by joint angles, velocities and accelerations and the contextual variables by the inertial parameters. To *acquire* the model, we need to have an estimate of $Y(q, \dot{q}, \ddot{q})$. If we have an appropriate number of models (that is, at least as many as the cardinality of π_r), we can simply estimate $Y(q, \dot{q}, \ddot{q})$ using least squares due to the linearity property. For *control* purposes, if we have an estimate of π_r at time t, given the desired transition for the next time step, we can compute $Y(q^*, \dot{q}^*, \ddot{q}^*)$ and hence, the feedforward command. For robust context estimation, we can use temporal dependencies, similar to the principles used in the multiple model scenario. However, since we now have a set of continuous hidden variables as opposed to a single discrete context variable, the inference is slightly more involved (refer to Table II).

But what does it mean for the quantity $Y_r(q, \dot{q}, \ddot{q})$ to remain constant in different contexts? Basically, it means that all kinematic quantities of the arm remain the same between different contexts – this is clearly true in the case of manipulating different objects.

Each link of the arm typically has ten inertial parameters. This implies that, ideally, if we have the prerequisite number of 'labeled' context models (for e.g., more than ten independent and perfect models corresponding to different scenarios), then, one can infer all the dynamic parameters of

any manipulated object. In practice, however, since learned dynamic models will not be perfect and due to the presence of noise in the sensor measurements, a larger number of 'context models' is necessary to give accurate estimates.

A. Experiments with the augmented model

The augmented model proposed for extracting the continuous context/latent variable was empirically evaluated. Separate models for the dynamics of the arm manipulating seven different objects with the *same shape* but different masses were trained and labeled. Masses were uniformly distributed between 0 and 0.06 where zero mass means load-free arm movement. Since all 10 inertial parameters of the manipulated object change linearly as the mass of the manipulated object changes between the contexts, just two known (labeled) contexts can be used to obtain the augmented model. While the scenario is less complicated than estimating the full moment of inertia matrix , successful estimation of the mass of the other five contexts and control using the augmented model can be used to validate the concept.

First, the accuracy of the augmented model was tested based on how well it can approximate other contexts' dynamics. We trained the augmented model using data from masses of 0.01 and 0.03. After parameter estimation, the learned model was used to predict the dynamic torques required to manipulate the other five contexts over a subset of the trained trajectory but for loads which have not been trained with. The error for the novel loads were computed by comparing
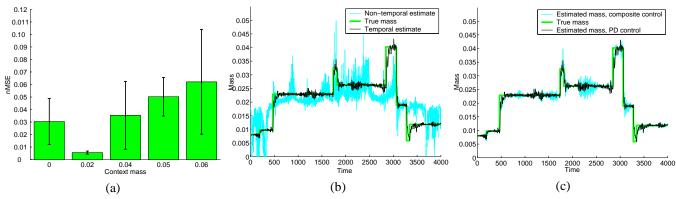
Fig. 8. Continuous context estimation: (a) Accuracy of the augmented inverse model (b) Temporal vs. Non temporal (c) Context estimates in control

the results of the augmented model with the torques learnt by the multiple, discrete models for the other five contexts. Fig. 8 (a) shows the nMSE of the torques predicted by the augmented model, averaged over all the joints. While the interpolated torques for the load of 0.02 is almost perfect, the extrapolated torques (outside the trained model of 0.01 and 0.03) for other loads also show excellent performance.

Next, we investigate whether the augmented model can be used for accurate context estimation. The augmented model (learned using the same two contexts) was used for context estimation under pure feedback control, where the mass of the manipulated object changed randomly during the simulation. Note that in this case, the context estimates were not used for computing the control commands. We compared the non-temporal and temporal formulation for context estimation, results for which are plotted in the Fig. 8 (b). The results for the non-temporal case are not that accurate and depend heavily on the choice of the prior of the context. However, when using the temporal model, the mass of the manipulated object is estimated quite accurately, irrespective of the prior. This can be explained by the fact that in the temporal case, the posterior of one time step effectively acts as the prior of the next, negating the influence of the initial priors beyond the start phase of the trajectory. Considering the real mass as the target, the nMSE of the estimates for the non-temporal and temporal cases were 0.8881 and 0.0423, respectively.

In the next step, we used the context estimates for control. The last simulation with continuous random changes in the context was repeated, but this time the arm was controlled with a composite controller and the augmented model was used to provide the feedforward command. Context estimates based on the temporal model were used as input to the augmented model. Fig. 8 (c) displays the accuracy of context estimates in this experiment compared to the one using pure feedback command. This comparison gives us an idea about how context estimation is affected when using the augmented model for control. Quantitatively, context estimation accuracy is a bit worse: in the previous experiment it was 0.0423 whereas it is now 0.0644. Furthermore, the efficiency of the augmented model in controlling the arm which can judged from the feedback to composite command ratio was 0.1495

– a result comparable to one we had for control with learned models under a single context.

V. DISCUSSION

We have described a method of using a learned set of models for control of a system with non-linear dynamics under *continuously* varying contexts. In addition , we have refined the multiple model paradigm to be able to *simultaneously* deal with learning dynamic models, use them for online switching control and also efficiently bootstrap data separation for context unlabeled data. An important aspect of this work is that we manage to infer the continuous hidden context that contains dynamic properties of the manipulated object, e.g. the mass of the object as illustrated in the experiments. While in this research, we have focused on estimating context purely from the predictive and experienced dynamics from manipulation, we are investigating avenues of incorporating a much richer sensory suite including haptic/tactile information from the hand to enhance context estimation and control.

REFERENCES

[1] John J. Craig. *Introduction to Robotics: Mechanics and Control.* Pearson Prentice Hall, 2005.

[2] Marc Toussaint Georgios Petkos and Sethu Vijayakumar. Learning multiple models of non-linear dynamics for control under varying contexts. In *Proceedings of International Conference on Artificial Neural Networks*, volume 1, pages 898–907, 2006.

[3] M. Haruno, D. M. Wolpert, and M. Kawato. Mosaic model for sensorimotor learning and control. *Neural Computation*, 13:2201–2220, 2001.

[4] R. A. Hilhorst and K. Tanie. Dexterous manipulation of objects with unknown parameters by robothands. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 4, pages 3098–3103. IEEE, 1994.

[5] K. S. Narendra and J. Balakrishnan. Adaptive control using multiple models. *IEEE Transactions in automatic control*, 42:171–187, 1997.

[6] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.

[7] M. K. Siliz and K. S. Narendra. Adaptive control of robotic manipulators using multiple models and switching. *International Journal of Robotics Research*, 1995.

[8] J.E. Slotine and W. Li. On the adaptive control of robot manipulators. *International Journal of Robotics Research*, 6(49–59), 1987.

[9] Sethu Vijayakumar, Aaron D'Souza, and Stefan Schaal. Incremental online learning in high dimensions. *Neural Computation*, 17:2602–2634, 2005.

[10] D. M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.