Hierarchical Dirichlet Process Hidden Markov Models for abnormality detection in robotic assembly

Enrico Di Lello, Tinne De Laet, Herman Bruyninckx
Department of Mechanical Engineering
Division Production engineering, Machine design and Automation
KU Leuven
Heverlee, BE 3001
first.last@mech.kuleuven.be

Abstract

The Hierarchical Dirichlet Process Hidden Markov model (HDP-HMM) is a Bayesian non parametric extension of the classical Hidden Markov Model (HMM) that allows to infer posterior probability over the cardinality of the hidden space, thus avoiding the necessity of cross-validation arising in standard EM training. This paper presents the application of Hierarchical Dirichlet Process Hidden Markov Models (HDP-HMM) to error detection during a robotic assembly task. Force sensor data is recorded for successful and failed task executions and manually labeled. An HDP-HMM is then fit to a set of training trials for each task execution outcome. We show how posteriors on the learned models could be used to recognize on-line deviation from expected behavior, thus allowing the robotic system to promptly react to task execution errors.

1 Introduction

Hidden Markov Models (HMMs) [1] have been applied for learning and recognition of time-series in fields that vary from speech recognition and genomics to robotics, computer vision, and finance. Over the decades, most of the theoretical advances have been focused on extending the original HMM expressive power by reformulating the HMM as a more general Dynamic Bayesian Network (DBN) leading to the development of layered, factorial, and hierarchical extensions (for a review, see [2]). While Expectation Maximization-based techniques provide valuable solutions to the problem of learning the parameters of an HMM, the one of selecting the cardinality of the hidden space (i.e. the model complexity) has been tackled either via cross-validation, model selection techniques developed in the parametric setting, or with *ad hoc* solutions.

Recently, a Bayesian non parametric extension of the classical HMM has been presented [3], named Hierarchical Dirichlet Process Hidden Markov model (HDP-HMM). It introduces the hierarchical Dirichlet process as a prior distribution on infinite dimensional transition matrices. In this way, the number of hidden states of the HMM becomes a variable of the model of which posterior densities can be computed via Gibbs sampling. The work of Fox [4] further extends the one of Teh by introducing an extra variable in the model that encourages transitions models with slower changes in the dynamics, therefore solving the problem the original HDP-HMM shows in favoring the learning of models with unrealistically fast-changing dynamics.

In this paper, we adopt this latter version, named by the author *sticky* HDP-HMM, to learn a set of *force sensor signature models* for a robotic assembly task using data recorded during a number of successful and faulty task executions. We then show how the learned models could be used to detect deviation from the expected force/torque data for a successful execution, therefore preventing and/or recognizing assembly errors.

An example of previous application of HMMs for monitoring robotic task execution is given in [5]. In their work, Howland *et. al*, focus on the detection of change in contact state between the workpiece and the environment. In [6], the focus is on assembly error detection, but a Support Vector Machine is used to classify force/torque measurement for successful and faulty executions.

To the best our knowledge, this paper represents the first attempt to apply the *sticky* HDP-HMM for learning a force signature model for a robotic assembly task.

Recent applications of HDP-HMM include speaker diarization [7], in which segmentation of audio recordings is required without knowing a-priori the number of speakers. In [8], HDP-HMM is used in combination of a one-class SVM classifier in the context of abnormal activity recognition. In [9] HDP-HMMs are used to synthesize audio clips of unlimited length after leaning the temporal and frequency pattern of a training set of recordings.

In our experiments, we assume to have a set of labelled force/torque time series, identified by an operator as arising from successful or faulty task executions. A subset of trials for a given execution outcome is selected for training and these trials are then jointly segmented using the *sticky* HDP-HMM to infer a model encoding the spatial and temporal correlations among the force/torque values of the trials. We then extract mean values from the learned posteriors over the HMM parameters, and use them to detect online deviations from the desired successful case (and possibly recognize the error type) using the standard forward-backward algorithm [1] in combination with a simple decision criterion.

The paper is organized as follows: Section 2 describes the robot set up and the assembly task, Section 3 details the application of the *sticky* HDP-HMM to the joint learning of a time-series model for a given execution outcome, Section 4 discusses the abnormality detection approach and its results, concluding in Section 5 with discussion of the results and ideas for future work.

2 Assembly Scenario

In this section we will shortly describe the robot setup used for recording the force/torque data and the assembly task. For a detailed description, we refer the reader to [10] and [11]. As we describe our particular robot setup and assembly task, we would like to stress the fact that the proposed approach is *neither task nor robot specific*. What our approach assumes is to have access to a number of force/torque measurement sequences recorded during successful and faulty task executions, labeled by a human expert. We also assume that the force/torque signatures recorded for different execution outcomes contain enough information to distinguish among them, while being to some extent consistent among executions with the same outcome. Given the fact that a typical robotic assembly task is usually performed following a well-defined sequence of sub-tasks (normally encoded in a Finite State Machine), and given the repeatability and accuracy performances guaranteed by and industrial manipulator, we believe this latter assumption is realistic.

2.1 Robot Setup and Assembly Task Description

The robot used for performing the assembly is FRIDA, the new concept robot from ABB [12], shown in figure 2.1. The assembly task performed is a subassembly of a mobile phone. A *shield can*, identified by the frame *f2* in figure 2.1 should be assembled onto a printed circuit board (PCB) mounted on a fixture, by applying pressure on a socket identified in 2.1 by *f1*. There are no mechanical tolerances between the parts, so the shield can will have to be deformed to fit.

The assembly strategy is designed such that the some of the uncertainty can be resolved in a robust way, and is divided in 9 sub-steps, encoded in a Finite State Machine (FSM).

Referring to figure 2.1, the assembly sequence can be described as follows:

- $step_1$ Pick up shield can from tray
- $step_2$ Go to start position
- $step_3$ Search for contact in negative *f1* $z-$direction
- $step_4$ Search for contact in positive *f1* $y-$direction
- $step_5$ Search for contact in negative *f1* $x-$direction
- $step_6$ Find corner of socket by yet another search in positive *f1* $y-$direction

$step_7$ Make a rotational search around the *f2* $x-$axis and the *f2* $y-$axis
$step_8$ Press shield can into position
$step_9$ Release shield can and move robot away

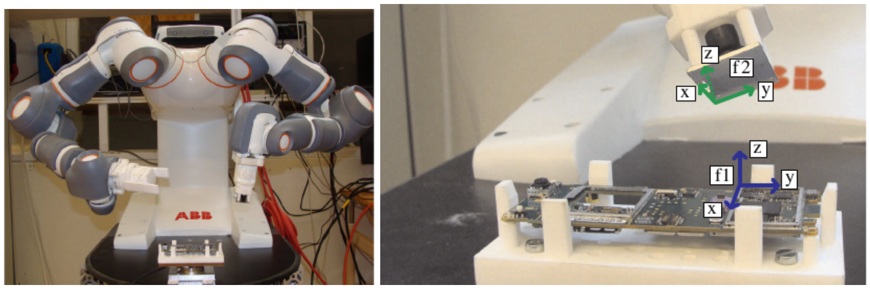


Figure 1: The ABB Frida robot [left] and assembly description: the shield can *f2* must be mounted on the PCB socket *f1* [right]

2.2 Dataset description

Data was recorded for a set of task executions for which possible 3 **execution outcomes** c_i were identified, namely:

- *success* : assembly task successfully completed (25 executions);
- $error_A$: shield not completely pressed onto the socket. This is caused by a not optimal grip of the shield can (2 executions).
- $error_B$: socket missed. This is caused by an error in the socket corner search (2 executions).

For each execution trial, the force/torque data is recorded from a sensor mounted beneath the PCB with a sampling time of $4ms$. While the assembly strategy briefly described in section 2.1 and implemented in [10] can to some extent deal with uncertainty and react to it, error in the assembly are still present. One of the reasons is that the FSM machine encoding the task is reacting to events based on force/torque *instantaneous thresholds*. Whenever a FSM state exit condition is verified (e.g. when a contact is detected in *f1* $z-$direction), the robot will blindly switch to the next step of the assembly, with no chance of discovering that an error might have occurred. Given that the *thresholds* triggering the robot FSM are fixed, we expect the *force/torque signatures* recorded during task execution to be different between the 3 execution outcomes, while being similar among trials with the same outcome. These expectations can be confirmed looking at figure 2, where the recorded force/torque data during one specific step of the assembly is shown for 2 of the possible outcomes. Each recorded trial can be automatically segmented looking at the task FSM evolution, thus obtaining the force/torque samples for each assembly sub-steps of Section 2.1. The end result of the data extraction process is a *set* of k (with k dependent on c_i) time-series of wrenches $\mathbf{w} = [f_x, f_y, f_x, \tau_x, \tau_y \tau_z]_k^T$ for each sub-step $s_j | j = 1:9$ and execution outcome $c_i | i = 1:3$.

3 HDP-HMM learning

This section provides a short overview of Dirichlet Process mixture model and its hierarchical extension. We will describe then how the latter can be used for learning the prior of the *sticky* HDP-HMM, and how we applied it to the learning of force/torque signature models for all the steps s_j of the assembly task described in 2.1 and for each possible execution outcome c_k. These models will be used in section 4 to develop a system for detecting on-line deviations from the nominal force/torque signature that leads to a successful execution, and possibly identify the error type.

3.1 DP,HDP and sticky HDP-HMM

The Dirichlet process (DP) is a distribution over probability measures on a parameter space Θ, uniquely defined by a *base measure* H on Θ and a *concentration parameter* γ, denoted by $DP(\gamma, H)$.

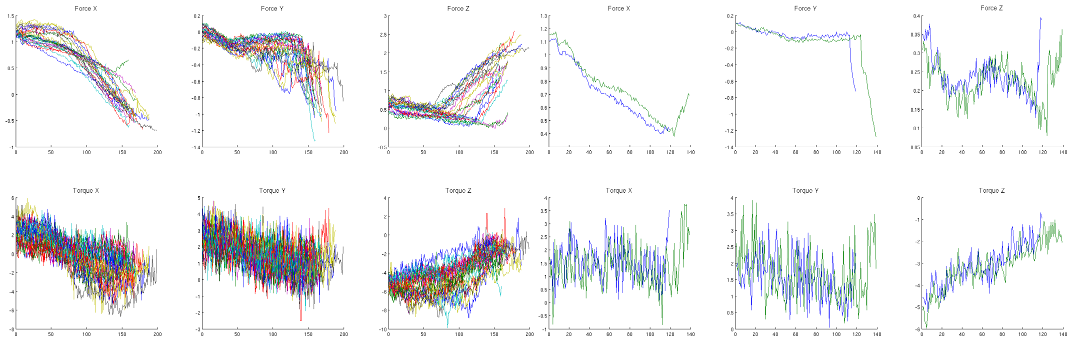


Figure 2: Force/Torque data of all the recorded executions for step 5 of the assembly task with outcome *success* (left) and outcome $error_A$ (right). First row: $[f_x, f_y, f_z]$ Second row: $[\tau_x, \tau_y, \tau_y]$. Differences can be noted between f_z and τ_x values for the different outcomes.

For space constraints, we will limit ourselves to a description of DP from the generative perspective, for more details, please refer to [3].

A random draw $G_o \sim DP(\gamma, H)$ can be expressed as

$$G_0 = \sum_{k=1}^{+\infty} \pi_k \delta_{\theta_k}(\theta) \quad \theta_k | H, \lambda \sim H(\lambda), \quad k = 1, 2, ... \tag{1}$$

where the $\delta_{\theta_k}(\theta)$ indicates a Dirac delta at $\theta = \theta_k$ and the weight $\beta_k \quad k = 1, 2, ; ...$ are obtained via a constructive procedure named *stick breaking construction* [13] denoted by $\beta \sim GEM(\gamma)$ and defined as following:

$$\beta_k \sim Beta(1, \gamma) \quad k = 1, 2, ... \tag{2}$$

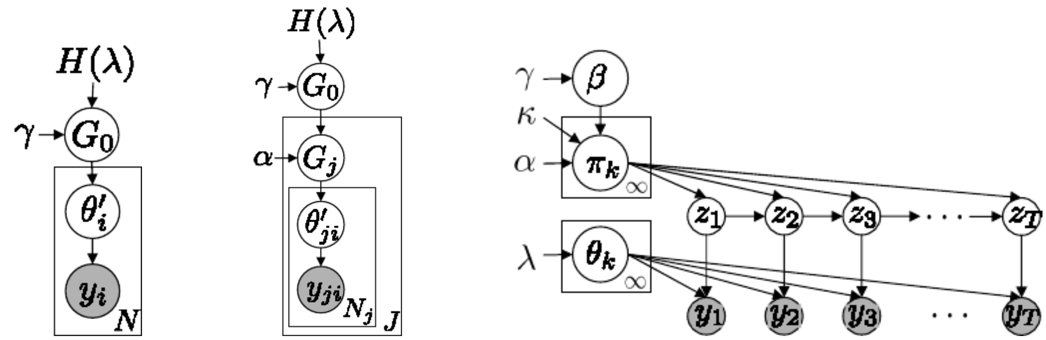
$$\pi_k = \beta_k \prod_{l=1}^{k-1} (1 - \beta_l) \quad k = 1, 2, ... \tag{3}$$

This constructive generative process can be interpreted as dividing a unit-length stick by the mixture weights π_k defined over an infinite set of random parameters θ_k drawn from the base measure $H(\lambda)$. The parameter γ controls the model complexity in terms of the expected numbers of components. If γ is unknown, a vague hyperprior can be placed on its value so that a posterior can be learned from data. Combining the DP prior with a likelihood distribution for the observations leads to mixture models with a potentially infinite number of components, graphically depicted in 3(a), where the θ_i' denote the parameter associated with the N observations y_i.

An interesting property of DP mixture models is that, since random probability measures drawn from a DP are discrete, there is a strictly positive probability that multiple observations will share a common parameter, leading to a natural *clustering* phenomenon. Furthermore, there is a *reinforcement* property that makes it more likely to associate observations to a parameter to which other observations have been already associated [7], therefore implicitly penalizing model complexity.

The *hierarchical* Bayesian extension of DP mixture is naturally derived if we assume that observations are produced by a related, yet distinct generative process. The hierarchical Dirichlet Process (HDP) [3] defines first a *global* base measure G_0, drawn by a $DP(\gamma, H)$ prior on the parameter space Θ, acting as an average distributions tying together the *group specific* distributions G_j, sampled from a $DP(\alpha, G_0)$. Since G_0 is discrete, the group specific distributions G_j will have overlapping support. This ensures that the mixture models in the different groups share mixture components. Figure 3(b) depicts the graphical model of the corresponding generative process, where now the θ_{ji}' denote the parameter associated with the N_j observations y_{ji} associated to group j (where the number of groups J is not known *a-priori*).

The HDP can be used to develop an HMM with an infinite state space, the HDP-HMM [3]. An HMM is a doubly stochastic process based on an underlying, discrete-valued state sequence modeled



(a) DP (b) HDP (c) HDP-HMM

Figure 3: Dirchlet process (left),hierarchical Dirichlet process (center) mixture models and HDP-HMM dynamic Bayesian network(right))

as markovian [1]. It is defined by an *initial transition distribution* π_0, a set of *state specific transition distributions* π_k, stacked into a *transition matrix* A, and a set of *emission parameters* θ_k, one for each state. In the HDP-HMM the *group-specific* distributions G_j of the HDP correspond to the *state-specific* transition distributions π_k, and due to the infinite state space, there potentially infinitely many of them, together with the corresponding emission parameters θ_k.

The properties of the HDP prior encourages different states to have a similar transition distributions π_k and to limit the number of (potentially infinite) states. Despite them, a problem with this model is that the HDP-HMM does not differentiate self-transitions from moves between different states, which leads to learned models with unrealistically fast dynamics, which can reduce the model predictive performance. To solve this issue, and be able to incorporate the knowledge that smoothly varying dynamics are more likely, Fox [4] introduced an extension of the standard HDP model, named sticky HDP, where an extra parameter κ is introduced that increases the expected probability of self-transition. The resulting generative model is given by:

$$\beta|\gamma \sim GEM(\gamma) \tag{4}$$

$$\pi_j|\alpha,\kappa,\beta, \sim DP(\alpha+\kappa, \frac{\alpha\beta+\kappa\delta_j}{\alpha+\kappa}) \quad j = 1,2... \tag{5}$$

$$\theta_j|H,\lambda \sim H(\lambda) \quad j = 1,2... \tag{6}$$

$$z_t|\{\pi_j\}_{j=1}^{+\infty}, z_{t-1} \sim \pi_{z_{t-1}} \quad t = 1,...,T \tag{7}$$

$$y_t|\{\theta_j\}_{j=1}^{+\infty}, z_t \sim F(\theta_{zt}) \quad t = 1,...,T \tag{8}$$

where β represent the vector of stick breaking weight for the first-level DP base measure, the π_j represent the second level DP measures corresponding to a state-specific transition distribution (i.e. a row of the HMM transition matrix), and the quantity $(\alpha\beta+\kappa\delta_j)$ indicates an amount $\kappa > 0$ that is added to the j^{th} component of $\alpha\beta$, representing the desired self-transition bias. When $\kappa = 0$, the original HDP-HMM of Teh [3] is recovered. Because positive values of κ increase the prior probability of self-transitions in the HMM, the model is referred as the *sticky* HDP-HMM. The variable z_t represents the HMM state index, therefore it is sampled from $\pi_{z_{t-1}}$ and indexes the parameter θ_{zt} used to generate observation y_t.

In figure 3(c) a Bayesian Network representation of the sticky HDP-HMM is given.

3.2 Learning a force/torque signature model using the sticky HDP-HMM

A MATLAB toolbox implementing several inference algorithms based on Gibbs sampling for the sticky HDP-HMM has been made available by Fox [14]. Gibbs sampling allows to compute the posterior distributions over all the parameters of the HDP-HMM in equations (4),(5),(6),(7),(8), conditioned on one or more sequences of observations, allowing to choose from a wide range of prior distributions. As our goal is to develop a system for abnormality detection, we used the blocked Gibbs sampler for the sticky HDP-HMM [4] to learn the posterior distributions over initial state probability π_0, the transition matrix A given by the collection of $\pi_1,...\pi_j$ and the corresponding set of emission parameters $\{\theta_1,...\theta_j\}$ capturing the time and spatial correlations from a *training set* composed of k wrench time-series $\mathbf{w} = [f_x, f_y, f_x, \tau_x, \tau_y \tau_z]_k^T$ for each sub-step s_j and execution outcome c_i.

Algorithm 3.2 contains the pseudo-code describing the adopted approach. For each task sub-step and for each execution outcome, we select randomly the 40% of the trials for training the model. We then iteratively apply the blocked Gibbs sampler for the HDP-HMM to the recorded wrench time-series. After convergence of the Gibbs sampler, the obtained samples approximate the posterior distributions for the HDP-HMM model variables (for a more detailed description of the adopted sampler and a review of Markov Chain Monte Carlo methods, see [4]). In particular, a set of sampled potential segmentations of the force/torque time-series corresponding to the state sequence samples $z_{t=1:T}$ of the HMM is obtained.

The approach proposed by Fox [4] is adopted for tackling the label-switching problem and choosing one representative segmentation out of the set of sampled ones, (for a detailed description of this phenomenon and a list of possible solutions, see [15]). At each iteration k, the Gibbs sampler is provided the observation sequence of the k time-series together with the selected state sequences for time-series $1:k-1$, so that both a combined set of observation parameters θ_k and the state sequence for time-series k are jointly inferred. This corresponds to informing the priors with the data from the $k-1$ time-series, and using the updated posterior distributions as the prior distributions for the time-series k [4].

---

for $i = 1:3$ **do**
- Pick randomly K training trials for execution outcome c_i
- $Train\_set = \{T_1^{c_i}, T_2^{c_i}, ...T_k^{c_i}\} \quad k = 1:K$
- **for** $s = 1:9$ **do**
  - $Segmented\_ts = \{\emptyset\}$
  - **for** $k = 1 \rightarrow K$ **do**
    - $HDP\_HMM\_samples$ = `HDP_HMM_Block_Sampler` $(Train\_set(k), Segmented\_ts)$
    - $ts\_k\_segmentation$ = `Find_Best_Segmentation`$(HDP\_HMM\_samples)$
    - $Segmented\_ts = Segmented\_ts \cup ts\_k\_segmentation$
  - **end for**
  - $step_{si}model$ =`Extract_Posterior_Means`$(HDP\_HMM\_samples)$
- **end for**

end for

For our experiments, we used 6-dimensional Gaussian emission parameters, and placed a weakly informative conjugate Normal Inverse Wishart (NIW) prior on the space of mean and variance parameters. The number of degrees of freedom was chosen as the minimum value necessary to obtain a proper prior, mean equal to the empirical mean of the observations, and scale matrix equal to 0.75 of the empirical variance. After convergence, the posterior means for the parameters $\Pi = \{\pi_0, A = \{\pi_1,...\pi_j\}, \Theta = \{\theta_1,...\theta_j\}\}$ are computed from the samples. Figure 3.2 shows an example of the learned segmentation for a time-series of $step_6$ of execution outcome c_1. The different colors correspond to different values of z_t and the horizontal lines represent the 2σ interval of the corresponding Gaussian emission parameter $\theta_{z_t} = \mathcal{N}(\mu, \sigma^2)$.

4 Abnormality detection and error recognition

The output of algorithm of section 3.2 consists of the estimated mean values for the parameters $\Pi_k = \{\pi_0, A, \Theta\}$, representing the initial state probability, the transition matrix and the set of learned Gaussian emission parameters $\theta_j = \mathcal{N}(\mu, \sigma^2)$, of an HMM capturing the time and spatial correlation among the training trials recorded for each task sub-step and each execution outcome.

The standard forward-backward algorithm [1] allows to compute the likelihood that a given number of observations were generated by any of the learned HMMs. For each of the validation time-series, we use the information stored in the FSM to identify the current $step_s$, and compute iteratively the likelihood $p(\mathbf{w}_{k:k+window\_size}|\Pi_{c_i(s)})$ for all classes c_i given the corresponding $step_s$ HMMs. We select a window size of 10 samples, and set a threshold on the *maximum time that any of HMM*

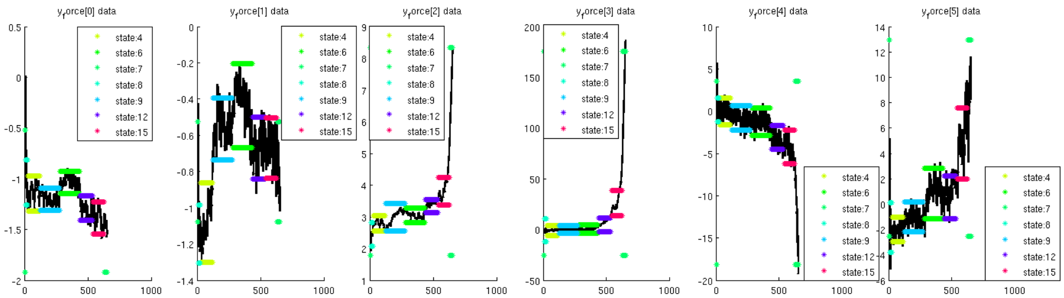


Figure 4: Learned segmentation of a training ts for $step_6$ of class *success*

corresponding to an error model is allowed to be the one best fitting the measurements. After this time, we consider the wrench data to be abnormal and identify the error as the one associated with the best fitting model. Figure 4 shows the evolution of the computed loglikelihoods during $step_6$ of a validation trial for $error_B$. The top plot represents the loglikelihood evolution, while the bottom one contains the same information but the loglikelihoods have been transformed in probabilities for better visualization. It can be seen how the probability of $error_B$, overcome the others between samples 150 and 250. Choosing a threshold of 75 samples or 300*ms*, we successfully identify abnormal observations, and we correctly identify the validation trials for $error_A$ and $error_B$, without causing any false positives for the 15 validation trials of class *success*. The same results are obtained after repeating 10 times the learning phase of Section 3.2 selecting randomly the training trials. The error recognition routines are currently implemented in MATLAB, but the proposed approach is suitable for an *on-line* implementation.

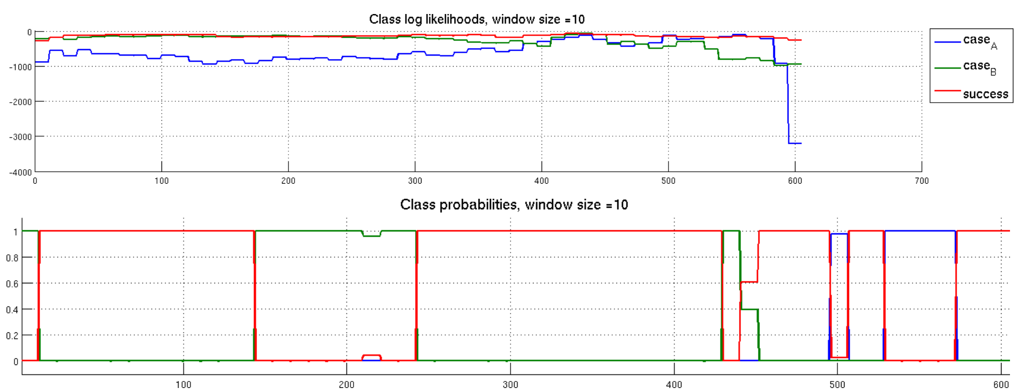


Figure 5: Likelihood and probabilities values for segment 6 of an execution of class $error_B$

5 Conclusion and Future Work

In this paper we have shown how HDP-HMMs can be used for learning *force/torque signature models* to recognize errors in robotic assembly tasks. Despite the limited data available for training and validation, we believe that the proposed approach is *neither task nor robot specific*, and similar performances can be obtained on richer datasets. We believe that two aspects in particular of the sticky HDP-HMM approach make it particularly interesting for other applications, namely the automatic model complexity selection, thanks to the sparsity induced by the HDP prior, and its fully Bayesian nature. This latter in particular is very interesting for robotics application, were expert and model-based knowledge about the expected signals could be modelled via informative priors. Future work include the application the proposed approach to a bigger dataset of different assembly tasks, and the investigation of the HDP-SLDS model [4], an extensions of the sticky HDP-HMM that allows the use of Linear Dynamical Systems as emission models. Research is currently ongoing on the application of this approach to human motion recognition.

Acknowledgments

The authors gratefully acknowledge the financial support by KU Leuven's Concerted Research Action GOA/2010/011, and the European FP7 project 2008-ICT-230902-ROSETTA. Tinne De Laet is a Post-Doctoral Fellow of the Fund for Scientific Research–Flanders (F.W.O.) in Belgium. We also thank Andreas Stolt, Magnus Linderoth and ULUND for providing us the assembly data.

References

[1] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[2] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, Computer Science Division, july 2002.

[3] Yee W. Teh, Michael Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

[4] Emily B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*. PhD thesis, Massachusets Institute of Technology, 2009.

[5] Geir E. Hovland and Brenan J. McCarragher. Hidden Markov Models as a Process Monitor in Robotic Assembly. *The International Journal of Robotics Research*, 17(2):153–168, Feb 1998.

[6] A. Rodriguez, D. Bourne, M. Mason, G. Rossano, and J. Wang. Failure detection in assembly: Force signature analysis. In *Proceedings of IEEE Conference on Automation Science and Engineering*, Toronto, Ontario, 2010.

[7] Emily B. Fox, Erik B. Sudderth, Michael I. Jordan, and Alan S. Willsky. An hdp-hmm for systems with state persistence. In *Proceedings of the International Conference on Machine Learning*, pages 312–319, 2008.

[8] Derek H. Hu, Zhang Xian-Xing, Jie Yin, Vincent W. Zheng, and Qiang Yang. Abnormal acitivity recognition based on hdp-hmm models. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence*, pages 1715–1720, Pasadena, California, 2009.

[9] Matthew D. Hoffmann, Perry R. Cook, and David M. Blei. Data-driven recomposition using the hierarchical dirichlet process hidden markov model. In *Proceedings of the International Computer Music Conference*, 2008.

[10] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson. Force controlled robotic assembly without a force sensor. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Minnesota, USA, 2012.

[11] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson. Force controlled robotic assembly of emergency stop button. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Shangai, China, 2011.

[12] S. Kock, T. Vittor, B. Matthias, H. Jerregard, M. Källman, I. Lundberg, R. Mellander, and M. Hedeling. Robot concept for scalable, flexible assembly automation: A technology study on harmless dual-armed robot. In *Int. Symp. Assembly and Manufactoring (ISAM)*, Tampere, Finland, 2011.

[13] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[14] Emily B. Fox. HDP-HMM toolbox. `http://www.stat.washington.edu/~ebfox/software.html`, 2009. Last visited 2012.

[15] A. Jasra, C.C. Holmes, and Stephens D.A. Markov chain monte carlo methods and and the label switching problem in bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005.