2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Bayesian Time-Series Models for Continuous Fault Detection and Recognition in Industrial Robotic Tasks

Enrico Di Lello, Markus Klotzbücher, Tinne De Laet and Herman Bruyninckx

Abstract— **This paper presents the application of a Bayesian nonparametric time-series model to process monitoring and fault classification for industrial robotic tasks. By means of an alignment task performed with a real robot, we show how the proposed approach allows to learn a set of sensor signature models encoding the spatial and temporal correlations among wrench measurements recorded during a number of successful task executions. Using these models, it is possible to detect continuously and on-line deviations from the expected sensor readings. Separate models are learned for a set of possible error scenarios involving a human modifying the workspace configuration. These non-nominal task executions are correctly detected and classified with an on-line algorithm, which opens the possibility for the development of error-specific recovery strategies. Our work is complementary to previous approaches in robotics, where process monitors based on probabilistic models, but limited to contact events, were developed for control purposes. Instead, in this paper we focus on capturing dynamic models of sensor signatures throughout the whole task, therefore allowing continuous monitoring and extending the system ability to interpret and react to errors.**

I. INTRODUCTION

The application of probabilistic models to industrial robotics has a few decades of history. Restricting ourselves to process monitoring and fault detection in industrial robotic tasks, which are the focus of this paper, we can summarize the relevant previous work in two main categories. The first category comprises approaches that focus on modelling events that *discretise* the task execution in sub-tasks relevant for control purposes. In [1], for example, Hidden Markov Models (HMMs) are used to model contact events between an object manipulated by a robot and the environment. By analysing the force/torque spectrogram, the type of contact can be identified among a discrete set of edge-surface configuration possibilities. Since a precise a priori model of both the shape of the object and the workspace is available, the recognition of the contact type can be used to trigger appropriate control actions that lead to the accomplishment of the task. Detecting errors in the task is possible, but limited to comparing the sequence of recognized contacts to the expected one. The error detection policy is therefore inherently discrete in nature, since the sensor measurements evolution *between* contact events is not modelled. In addition, the success of the task execution is based on the assumption that there is a well known correspondence between contact events and the robot "world" state (i.e. relative positions between the robot itself, the manipulated object and the workspace). As will be shown later, these assumptions is restrictive if we consider a human-robot cooperation scenario, because the human might, voluntarily or not, introduce deviations from the expected world state that cannot be immediately detected by the robot. In this scenario, the robot should be able to *continuously* detect these deviations, possibly recognize them, and act accordingly.
In the second category fall those approaches that have the interpretation of some sort of measurement time-series as a key feature. In [2], Support Vector Machines are used in combination of Principal Component Analysis, for classification of force/torque measurement time-series. Under the assumption that the wrenches time-series recorded during the task are correlated with the execution outcome, successful task execution can be discriminated from faulty one. While the adopted classification method is general purpose and model-based, task-specific knowledge is not required, the error detection can only happen upon task completion. In [3] the authors overcome this limitation, applying Relevance Vector Machines in combination with a Markov Chain model, to continuously estimate the quality of a grasp from the gripper joint angles time-series. In this way, the robot can detect on-line whether to continue the task or abort it to perform a better grasp. Another example of wrench time-series analysis for assembly outcome verification can be found in [4], where a hierarchical taxonomy is built to classify the outcome of a snap assembly from a recorded wrench signature. The development of such a taxonomy is done by visual exploration of sequences of linear segment primitives in the time-series, and can hardly be re-used for a different assembly. Furthermore, the proposed approach is validated only on simulated data.

In this paper, we focus on Bayesian time-series model that can be applied to heterogeneous sensors, allowing continuous and on-line detection of deviation from nominal executions of an industrial robotic task. This is obtained by learning sensor signatures throughout the whole task and not limiting ourselves to contact events. The proposed approach requires little or none model-based, task specific knowledge, but still allows its integration in the learning process if available. The assumptions on which the proposed method relies can be summarized as follows:

1) during a number of task execution, data is recorded from one or more sensors (e.g. force/torque, joint encoders).

The authors are with the Department of Mechanical Engineering, Division Production engineering, Machine design and Automation, KU Leuven, 3001 Heverlee, Belgium

2) the task execution scenario (and/or its outcome) can be identified by a human expert, and the corresponding recorded data is labelled accordingly (e.g. *nominal* or *error_scenario_1*). If only deviations from the nominal task execution must be detected, than data should be recorded only for *nominal* execution trials. If error *classification* is required, example trials for each error scenario must be recorded as well.
3) the recorded sensor signature is correlated with the task execution scenario (and/or its outcome). That is, the sensor time-series (signature) contains, at some point in time during the task, enough information to detect deviations from the expected measurements.

In this paper, we show that the proposed method can successfully be applied to learning force/torque sensor signatures for of a simple alignment task. The learned models can be used on-line to detect deviations from nominal task executions and classify the error cause between a set of previously identified scenarios.
While the task presented in this paper is not challenging *per se*, the wide applicability of the proposed method is demonstrated by our preliminary work [5], where the assembly task, the robotic setup and the sensor setup where different, but the aforementioned assumptions where still satisfied.

The paper is organized as follows: in Section II the robotic setup, the example task, and a set of possible non-nominal execution scenarios arising from human interaction, are described. In Section III the adopted time-series model for learning the sensor signatures is described. In Section IV we explain the on-line error recognition approach and the performance for real executions of the example task are examined. In Section V we conclude by describing other potential application scenarios and discuss future directions.

II. A ROBOTIC ALIGNMENT TASK UNDER CHANGING WORKSPACE CONDITIONS

In this Section we will describe an example robotic task and how the robot and the developed control strategy behaves under unpredictable changes in the workspace due to human intervention. The focus is neither on the complexity of the task nor on the optimality of the execution strategy. This example task serves as an example of a generic robotic task whose nominal execution can be defined via a Finite State Machine (FSM) and executed via a hybrid control strategy, like for example in [6]. The FSM discretises the task in a set of intermediate actions (sub-tasks) and defines for each of those a specific control strategy and an "event" that triggers the transition to the next state. These events are usually defined by *thresholds* on measurable quantities (e.g. a force or a torque) and implicitly define a correspondence between them and a specific world state. This means that during the sub-tasks design phase, these events are usually defined by a human expert via trial-and-error, with the task executed under constant conditions. Force/torque events are usually linked to contact events, and sequence of contact events are assumed to correspond to a desired configuration of the manipulated object, the robot, and the workspace. To demonstrate the limitations of this approach we describe a simple alignment task where an object (specifically, the case of a stop button) must be aligned to one side of a rectangular workspace. The task execution strategy is encoded in a FSM, where events are defined on force/torque measurement, time, position, or joint angles and is successfully executed. We then introduce an unmodelled object in the robot workspace, that interferes with the nominal task execution. This scenario can naturally arise in the context of human-robot cooperation, where part of an assembly is performed by a human, and a semi-assembled part is involuntarily dropped or delivered to the robot in a position different than expected. We will show how this change in the workspace configuration breaks the correspondence between contact events and the assumed world state, leading to unpredictable task executions.

A. Robot setup and Task FSM

The robot setup used for the example alignment task is a KUKA LWR. Wrench measurement can be obtained via the Fast Research Interface [7], and recorded during task execution at $500Hz$. We briefly describe the FSM developed to perform the task, as well as the transition events. The goal of the task is to align the yellow stop button case to one side of the table (see figure 1). To achieve this, the task is decomposed in the following sub-tasks, encoded together with their transition events in a FSM:

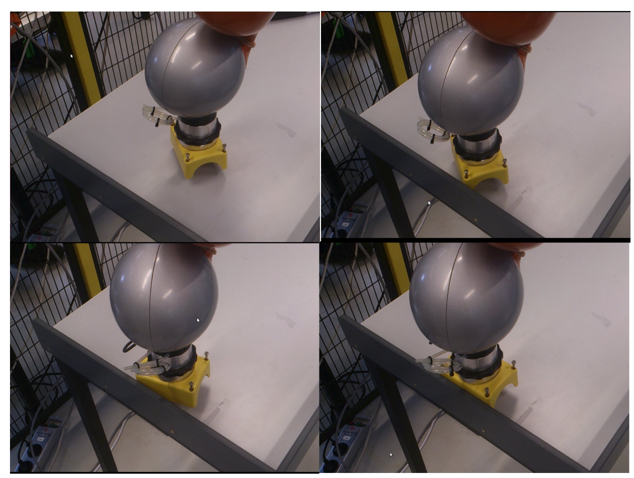


Fig. 1. The example alignment task: after $step_1$(top-right), after $step_2$(top-left), after $step_3$(bottom-left), after $step_6$(bottom-right)

$step_1$ Rotate the case by 45° around the z axis. When the task is started, the case is assumed to be roughly pre-aligned, therefore this motion results in one corner of the case pointing approx. towards the expected contact point.

$step_2$ Move towards the table until contact is detected (threshold on $y-axis$ force, $-7N$).

$step_3$ Rotate the case around z-axis until a new contact is detected. This should mean that opposite corner has made contact with the table. (threshold on z-axis torque, $0.5Nm$). The duration of this step is measured.

$step_4$ Rotate around z in the opposite direction of $step_3$ for the previously measured time divided by two. After this, the case should be reasonably well aligned with respect to the table.

$step_5$ Move towards the table as in $step_2$.

$step_6$ Upon contact, align in y direction, by applying a force of $-20N$ in y-direction for 3 seconds, while being compliant in the rotational d.o.f x and y.

$step_7$ Move back

A video of a nominal and a faulty execution of the task can be found at `http://people.mech.kuleuven.be/~edilello`. Additional details about implementation and control strategies are beyond the scope of this paper.

B. Nominal and abnormal task executions

When implementing a FSM for the alignment task, the system designer implicitly defines a correspondence between the measured wrenches and a desired intermediate configurations of the robot world that are supposed to lead to the achievement of the task. This correspondence can be broken if an undesired object is accidentally introduced in the workspace. As an example, consider the scenario depicted in the figure 2, where different objects were placed in front of the table against which the stop button case is supposed to be aligned. It can be foreseen that the robot will detect a contact when touching the object, and as long as the event-specific thresholds are crossed, the FSM will switch to the corresponding states. The results in terms of the accomplishment of the alignment are unpredictable, but the outcome might be a graceful fail as well as a damaging of the robot or the manipulated objects. Clearly, this is true for our particular implementation of this task. The robustness of this task could be improved, for example, by extending the FSM introducing intermediate states whose goal is to confirm that contact has been made as expected.

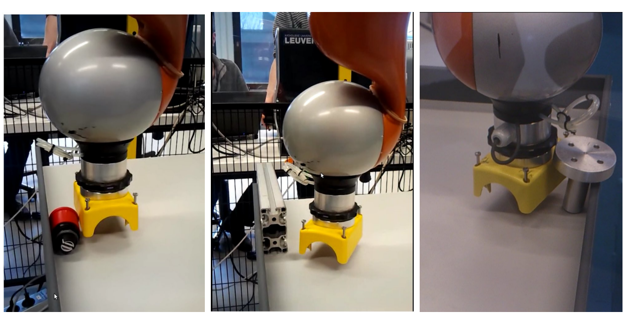


Fig. 2. Alignment task execution with unmodelled obstacles: *button* (left), *metal square* (center), *metal round*(right)

What we want to highlight is that this kind of mismatches between a detected event and the assumed world state can be solved by focusing on how some measurable quantity evolves over time and linking this *time-series* to a semantically richer and probabilistic description of the world state (i.e. "sub-task succeeded with probability 0.9" instead of "ForceX above threshold"), allowing for continuous error detection and interpretation.

Data was recorded for a 24 task execution in 5 different scenarios $class_c$, $c \in \{1,\ldots,5\}$:

- *nominal* : clear workspace, task successfully completed (8 executions);
- *metal round* : round metal object placed between the robot and the table (5 executions);
- *metal square* : square metal object placed between the robot and the table (4 executions);
- *button* : round plastic stop button placed between the robot and the table (3 executions);
- *box* : stop button case placed between the robot and the table (4 executions);

In all the non-nominal execution scenarios, the robot went trough all the FSM states, but did not align successfully with the table, because it was unable to correctly estimate the desired angle due to contact with the unexpected objects. Depending on the execution scenarios, the wrench signature will diverge, at some point of the task, from the nominal ones. As an example, we show the recorded wrenches for $step_4$ of the FSM, for all $class_c$, $c \in \{1,\ldots,5\}$ in figure 3. It can be seen that the sensor signatures are different among execution scenarios while still being somewhat consistent among trials of a given scenario.

III. LEARNING A BAYESIAN TIME-SERIES MODEL OF WRENCH SIGNATURES

This section provides a short overview of the Bayesian time-series model named *sticky*-Hierarchical Dirichlet Process Hidden Markov Model (*sticky*-HDP-HMM) [8], used for learning force/torque signature models for all the steps $step_s$ of the example alignment task described in II-A and for each possible execution outcome $class_c$. The learned models will be used in section IV to develop a system for detecting on-line deviations from the nominal force/torque signature that characterize a successful task execution, and possibly identify the specific execution scenario $class_c$. Depending on the task, deviation from the nominal task execution might be detectable by different sensors and only from a particular point in time during task execution. Therefore, it is necessary to:

- segment the recorded time-series according to the FSM evolution;
- select the specific sensor that allows for the deviation to be detected;
- label the corresponding time-series according to the task execution scenario $class_c$ (i.e. *nominal*, *metal part*).

While the first is done automatically we assume that the second and the third steps are performed manually. Approaches to automate these latter two steps are found in literature, but are beyond the scope of this paper.

A. Hidden Markov Models

The *Hidden Markov Model* is a statistical signal model widely used for speech recognition, natural language modelling, on-line handwriting recognition, and the analysis of biological sequences such as proteins and DNA [9]. In

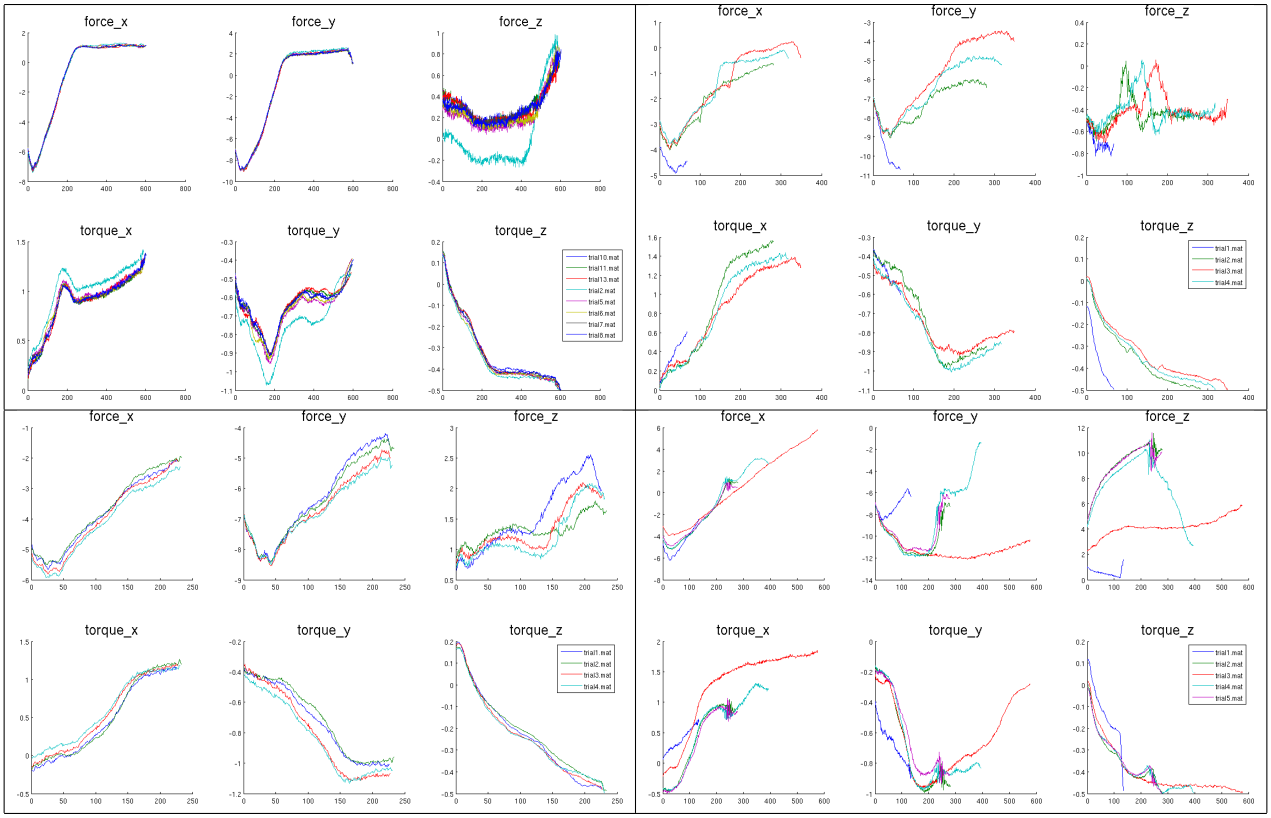


Fig. 3. Wrenches recorded during $step_4$ of execution scenario *nominal* (top-left), *metal square* (top-right), *button*(bottom-left), *metal round*(bottom-right)

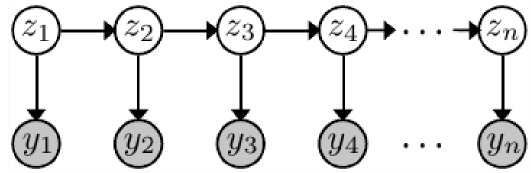
robotics, HMMs and their extensions have been successfully applied to encode human motion primitives and transfer them to a humanoid robot [10], to programming by demonstration [11], and to model contact events in assembly tasks [1] . The HMM can be seen as a stochastic finite state automaton, where each state emits an observation. More in detail, at each time t the observations Y_t are considered a probabilistic function of the state Z_t . The state Z_t is represented by a discrete random variable with $Z \in \{1 \dots K\}$, where K is the *cardinality* of the hidden state, evolving according to a stochastic process that is not observable (i.e. *hidden*), and can only be observed through the produced sequence of observations [12]. A HMM is represented as a Dynamic Bayesian Network [13] (DBN) in figure 4(a) and is defined by the following set Π of model parameters:

- an initial state distribution $\pi_0 = [\pi_1, \pi_2, \dots, \pi_n]$, where $\pi(i) = P(Z_1 = i)$. π_0 is represented by a multinomial distribution.
- a transition model, represented by the stochastic matrix A where $A(i, j) = P(Z_t = i|Z_{t-1} = j)$, modelling the evolution of the unobservable discrete state. Each row of the matrix defines a conditional multinomial distribution.
- a set Θ of observation models, defining the probabilities $P(Y_t = y|Z_t = i)$. If the observations are defined as continuous feature vectors then a conditional continuous p.d.f., (usually Gaussian) is defined as the emission distribution for each state $\Theta = \{\theta_1, \dots \theta_K\}$.

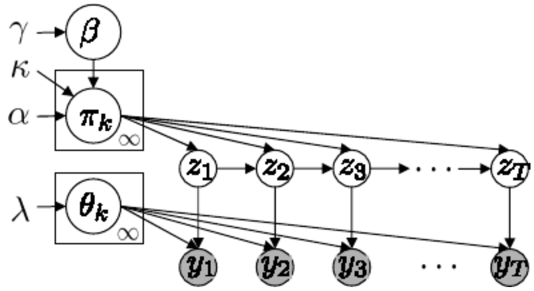
In our experiments, we used a HMM with Gaussian observation models to capture the spatial and temporal correlations among the measured wrenches and their first order differences for each sub-task encoded in the FSM, and for each different execution scenarios. The choice of extending the wrench measurements with their first order derivatives is motivated by better recognition performances. The idea is that the segmentation of the time-series induced by the HMM is influenced by changes in both *location* and *slope* of the signals. What we obtained is a set of probabilistic models that can represent the different sensor signatures shown in figure 3 and be used for error detection and classification.

B. Bayesian Nonparametric Learning of HMMs: the sticky-Hierachical Dirichlet Process

In robotics, the problem of estimating the parameters of a HMM has mostly being solved via Expectation-Maximization (EM) algorithm [9], which allow estimate the HMM parameters that maximize the likelihood of the observations contained in a set of training time-series. An alternative approach is given by Bayesian methods, in which prior distributions are placed on the HMM parameters themselves, and then a *maximum a posteriori* estimate of these parameters, conditioned on a training set, is obtained. Recently, a Bayesian nonparametric method for learning HMM with an infinite state space, namely the Hierarchical Dirichlet Process (HDP) - HMM has been proposed [14]. The HDP is a distribution over probability measures on a parameter space



(a) HMM



(b) HDP-HMM

Fig. 4. Hidden Markov Model (4(a)), HDP-HMM dynamic Bayesian network(4(b))

Θ, that encourages different states to have similar transition distributions (defined by the rows of the transition matrix A) and to limit the number of (potentially infinite) states. The HDP-HMM has been extended by Fox [15] to bias the posterior distributions towards models with smoothly varying dynamics, and has therefore being named *sticky*-HDP-HMM, whose DBN is shown in 4(b). In this paper, we used this latter approach to learn wrench signature models, using the Gibbs sampling routines made available by Fox [16].

For space reason, we will not discuss the details of the sticky-HDP-HMM. For the particular application on which this paper focuses, the advantages of a fully Bayesian approach over EM are manifold. The definition of prior distributions on the HMM parameters, opens the possibility to integrate domain-specific knowledge about the static and dynamic features of the sensor signature that we are trying to fit. For example, referring to figure 4(b), the hyper-parameter γ can influence the number of unique observation models generated to fit the time-series observations, κ can bias the self-transition probabilities of the transition matrix A, while λ can be used to define a priori knowledge about the spatial distribution of the observations (in case of a Gaussian observation models). If this knowledge is not available, *weakly informative priors* can be used. Furthermore, the HDP-HMM can naturally be extended to a model were the observation models consist of Linear Dynamical Systems [17], allowing to take into account the correlations between measurements and therefore represent a more natural choice for smooth trajectories. Finally, the nonparametric treatment of the hidden space cardinality eliminates the need for cross-validating this parameter that arises in Maximum Likelihood settings (i.e., when using the EM algorithm).

We used the blocked Gibbs sampler for the sticky HDP-HMM [15] to learn the posterior distributions over the initial state probability π_0, the transition matrix A, 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, together with their first order derivatives $\mathbf{z} = \left[f_x, f_y, f_x, \tau_x, \tau_y \tau_z, \dot{f}_x, \dot{f}_y, \dot{f}_x, \dot{\tau}_x, \dot{\tau}_y, \dot{\tau}_z,\right]_k^T$ for each sub-step $step_s$ and execution scenario $class_c$.

For each task sub-step and for each execution scenario, we randomly select 50% of the trials for training the model. We then 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 [15]). For our experiments, we used 12-dimensional Gaussian emission parameters, and placed a weakly informative conjugate Normal Inverse Wishart (NIW) prior on the space of mean and variance parameters (we therefore assume full correlation between the dimensions of $\mathbf{z}$). The number of degrees of freedom was chosen as the minimum number necessary to obtain a proper prior, the mean equal to the empirical mean of the observations, and the scale matrix equal to 0.75 of the empirical variance. After convergence, the posterior means for the parameters $\Pi = \{\pi_0, A, \Theta = \{\theta_1, ...\theta_j\}\}$ are computed from the samples. Figure 5 shows an example of the learned segmentation for a time-series of $step_4$ of execution scenario *nominal*. The different colours correspond to different values of z_t and the horizontal lines represent the 2σ interval of the corresponding Gaussian emission parameters $\theta_{z_t} = \mathcal{N}(\mu, \sigma^2)$.

IV. TASK EXECUTION MONITORING AND ABNORMALITY CLASSIFICATION

The output of the Gibbs sampler of section III-B 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 a HMM capturing an "average" wrench signature of the training trials recorded for each task sub-step and each execution scenario.

The standard forward-backward (fwd-bwd) algorithm [12] allows to compute the likelihood that a sequence of observations was generated by a learned HMM.

For each of the 11 task executions not used for training, we perform online error detection and classification as follows:

- iterating over the wrench time-series, we identify the current $step_s$ from the FSM;
- in case only error detection is required, we select only the learned HMMs $nominal_s$ and run the fwd-bwd algorithm to iteratively compute the likelihood $p(\mathbf{z}_{s_1:s_t}|\Pi_{nominal_s})$, where t goes from the first to the last time-index where the FSM in step s. If error classification is required the same is done for the HMMs $class_{cs}$ learned for all the execution scenarios $class_c$.
- again, if only error detection is required it is sufficient to raise an error signal when the $p(\mathbf{w}_{s_1:s_T}|\Pi_{nominal,s})$ falls below a threshold, determined via ROC analysis [18]. In case of error recognition the time-series is

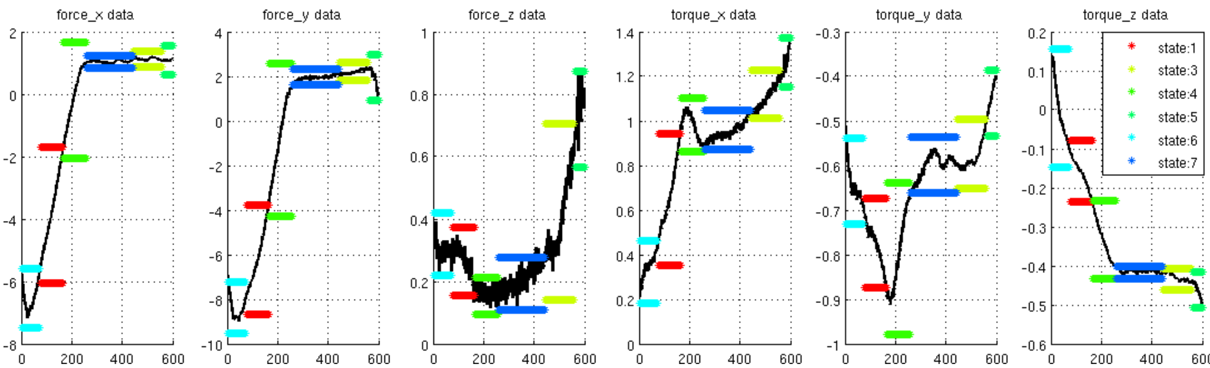


Fig. 5. Learned segmentation for a wrench time-series of $step_4$ of execution scenario *nominal*. The different colours correspond to different values of z_t and the horizontal lines represent the 2σ interval of the corresponding Gaussian emission parameters. Plots for wrench first order derivatives are omitted.

assigned to the $class_c$ for which the computed likelihood is maximal.

A. Error Detection and Recognition: experimental results

The error detection and classification recognition routines are currently implemented in both MATLAB scripting language and in C++, allowing their integration with the robot controller while preserving real-time performances. The whole learning procedure of section III was repeated 10 times, each time randomly selecting the execution trials used for training. Error detection and recognition is performed on the remaining 11 execution trials. The earliest sub-task for which the sensor signature allow for reliable error detection is $step_4$. Figure 6 shows the ROC curve for $step_4$ obtained averaging over the 10 learning/classification iterations. A ROC curve illustrates the performance of a binary classifier system as its discrimination threshold is varied. In our case, we use the log-likelihood $\log(p(\mathbf{w}_{s_1:s_t}|\Pi_{nominal_s}))$ to discriminate between sensor signatures generated by trials belonging to the *nominal* execution scenario, and *any of the others*. The x-axis refers to the *false positive rate* (i.e. the percentage of non *nominal* trials erroneously classified as *nominal*, divided by the total number of alarms raised), while the y-axis to the true positive rate. The curves can be interpreted as follows: the ROC curve for $step_4$, indicates a minimum true positive rate of 97.5 . This is consistent with the confusion matrix shown in table I, where it is shown that in only one case an execution trial of scenario *nominal* is misclassifed as *button*. This confirms our hypothesis that under the assumptions mentioned in section I the learned HMMs can successfully model the difference sensor signatures and that this models can be used for on-line error detection. In terms of error classification, table I shows the confusion matrix for the classification of all the validation trials (for space reason, we restrict ourselves to the time-series of $step_4$), averaged over the 10 learning/recognition iterations. On average, 91% of the validation trials are correctly classified, with a maximum of 97.5% for class *nominal*, and a minumum of 80% for class *box*. In figure 7, we show how the class-specific log-likelihoods for each measurement w_t and for measurements $w_{s_1:s_t}$ evolve as more measurements $\mathbf{w}_t$, recorded during

$class_{c4}$	$nominal_4$	$metal\ round_4$	$metal\ square_4$	$button_4$	box_4
$nominal_4$	**97.5%(39/40)**	0	0	2.5%	0
$metal\ round_4$	0	**90%(18/20)**	10%	0	0
$metal\ square_4$	10%	0	**90%(18/20)**	0	10%
$button_4$	0	0	10%	**90%(9/10)**	0
box_4	0	0	20%	0	**80%(16/20)**

TABLE I

RECOGNITION PERFORMANCES FOR 11 TRIAL RANDOMLY SELECTED VALIDATION TRIALS FOR ALL EXECUTION SCENARIOS, FSM $step_4$, AVERAGED OVER 10 LEARNING/RECOGNITION ITERATIONS

$step_4$ of a *nominal* execution trial, are examined. It can be noted how, for $t < 325$ (where t is a time-step index, sampling rate is $500Hz$), the likelihood of class *metal round* is the highest, with *nominal* being the second most-likely. For $t > 325$, the cumulative likelihood of the measurements $\mathbf{w}_{1:t}$ are estimated to belong to the *nominal* scenario. The classification results shown in this section refer to the case where a decision is made at the end of any $step_s$. Hastening the decision time allows for faster reactions, but is more likely to cause false alarms.

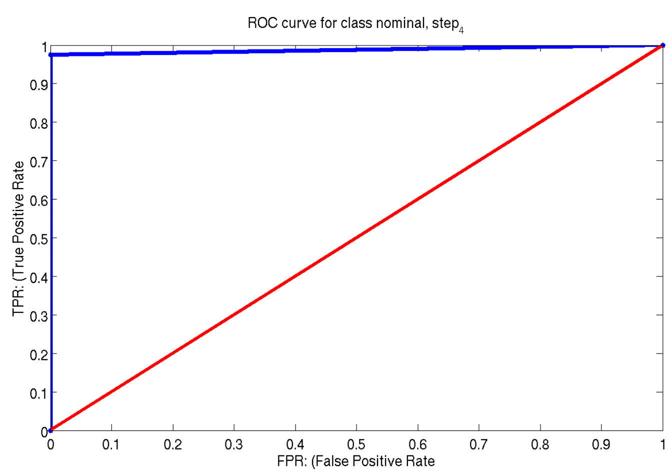


Fig. 6. ROC curves (for the 10 learning/recognition iterations) for $step_4$, obtained varying the threshold for $\log p(\mathbf{w}_{s_1:s_t}|\Pi_{nominal_s}$ in red the no-discrimination line

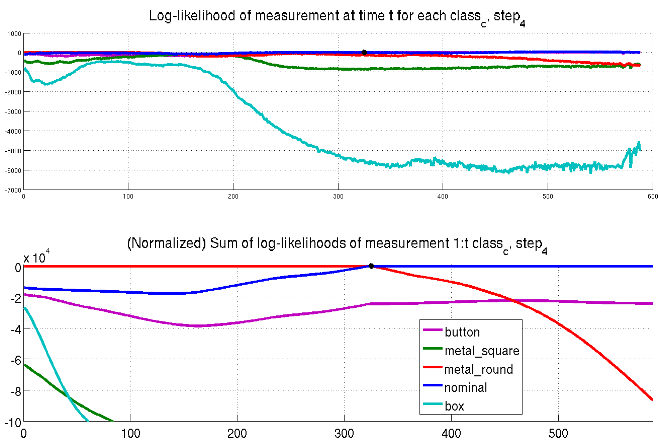


Fig. 7. (top) Plot of $\log p(\mathbf{w}_{s_t}|\Pi_{class_{cs}})$ and $\log p(\mathbf{w}_{s_1:s_t}|\Pi_{class_{cs}})$ (bottom) for $step_4$ of a *nominal* execution trial. From $t > 325$, the log-likelihood of all the measurements from s_1 to s_t is constantly highest for the *nominal* class.

V. CONCLUSION AND FUTURE WORK

In this paper we have shown how a Bayesian time-series model, namely the *sticky*-HDP-HMM, can be applied for process monitoring and fault classification in industrial robotics tasks. By means of an alignment task performed with a real robot, we demonstrated the ability of the *sticky*-HPD-HMM to capture sensor signatures for a nominal and abnormal task executions. These signature models are then used on-line to perform error detection and classification. The proposed method is widely applicable to different tasks and sensors, since its assumptions are limited to the availability of human-labelled task execution data, and the amount and quality of the information content in the sensor signatures. The relevance of this work for the field lies in the possibility to extend the semantics of *events* usually used for modelling task FSM. We envision high-level robot controllers that can use the typical *discrete* events used in sub-task modelling, and at the same time measure *continuously* and *probabilistically* the correspondence of sensors *time-series* to the expected ones, when this knowledge is available. The possibility of identifying and react to semantically richer error events is of great importance in context of human-robot cooperation scenarios, where the assumptions of precise modelling of the workspace and its static nature have to be relaxed. An interesting future work direction consists in the investigating the use of the *sticky*-HDP-VAR and HDP-SLDS models [15], extensions of the sticky HDP-HMM that allows the use of Vector Auto-Regressive (VAR) and Linear Dynamical Systems (LDS) as observation models. These models assume correlation among the measurements associated to a given state z, and are therefore better suited to model smooth signals like the ones recorded in this setting. Unfortunately, fitting these model is computationally more expensive in both the learning and the recognition steps, and require more informative priors to obtain reliable posterior models. These priors could be obtained leveraging available expert knowledge about the expected dynamic content of the sensor signatures.

ACKNOWLEDGMENT

The authors gratefully acknowledge the financial support by the European FP7 projects ICT-230902-ROSETTA, ICT-231940-BRICS, OT/12/100, and KU Leuven's Concerted Research Action GOA/2010/011)

Tinne De Laet is a Postdoctoral Fellow of the Fund for Scientific Research–Flanders (F.W.O.) in Belgium.

REFERENCES

[1] G. E. Hovland and B. J. McCarragher, "Hidden Markov Models as a Process Monitor in Robotic Assembly," *The International Journal of Robotics Research*, vol. 17, pp. 153–168, Feb 1998.

[2] 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.

[3] A. Rodriguez, M. Mason, S. Srinivasa, M. Bernstein, and A. Zirbel, "Abort and retry in grasping," in *Proceedings of IEEE Conference on Intelligent Robots and Systems*, (San Francisco, CA, USA), 2011.

[4] J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida, K. Nagata, and Y. Kawai, "A relative-change based hierarchical taxonomy for cantilever-snap assembly verification," in *Proceedings of the IEEE International Conference on Robots and System (IROS)*, pp. 356–363, 2012.

[5] E. Di Lello, T. De Laet, and H. Bruyninckx, "Hierarchical dirichlet process hidden markov models for abnormality detection in robotic assembly," in *Workshop on Bayesian Nonparametric Models (BNPM) For Reliable Planning And Decision-Making Under Uncertainty, NIPS 2012*, 2012.

[6] 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.

[7] G. Schreiber, A. Stemmer, and R. Bischoff, "The Fast Research Interface for the KUKA Lightweight Robot." In IEEE Workshop on Innovative Robot Control Architectures for Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010), May 2010.

[8] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An hdp-hmm for systems with state persistence," in *Proceedings of the International Conference on Machine Learning*, pp. 312–319, 2008.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[10] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive Hidden Markov Chains," vol. 27, no. 7, pp. 761–784, 2008.

[11] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*. EPFL/CRC Press, 2009.

[12] L. R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

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

[14] Y. W. Teh, M. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, pp. 1566–1581, 2006.

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

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

[17] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Bayesian nonparametric methods for learning markov switching processes," *IEEE Signal Processing Magazine*, 2010.

[18] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006.