Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 23 (2014) 229 – 234

www.elsevier.com/locate/procedia

Conference on Assembly Technologies and Systems

Contact-State Modeling of Robotic Assembly Tasks Using Gaussian Mixture Models

Ibrahim F. Jasim[a,*], Peter W. Plapper[a]

[a]*Faculty of Science, Technology, and Communication, University of Luxembourg, L-1359 Luxembourg*

* Corresponding author. Tel.: +352-4666-44-5353; fax: +352-4666-44-5200. *E-mail address:* ibrahim.jasim@uni.lu

Abstract

This article addresses the Contact-State (CS) modeling problem for the force-controlled robotic peg-in-hole assembly tasks. The wrench (Cartesian forces and torques) and pose (Cartesian position and orientation) signals, of the manipulated object, are captured for different phases of the robotic assembly task. Those signals are utilized in building a CS model for each phase. Gaussian Mixture Models (GMM) is employed in building the likelihood of each signal and Expectation Maximization (EM) is used in finding the GMM parameters. Experiments are performed on a KUKA Lightweight Robot (LWR) doing camshaft caps assembly of an automotive powertrain. Comparisons are also performed with the available assembly modeling schemes, and the superiority of the EM-GMM scheme is shown with a reduced computational time.

Selection and peer-review under responsibility of the International Scientific Committee of 5th CATS 2014 in the person of the Conference Chair Prof. Dr. Matthias Putz matthias.putz@iwu.fraunhofer.de

Keywords: Contact-State modeling; force-controlled robots; gaussian mixture models; robotic assembly;

1. Introduction

Assembly is considered one of the vital topics for both industry and research institutions and automating the assembly for different products drew the attention of many practitioners from both academia and production sectors. Robots are considered the most important tools in automating productions and hence researches in this field are considered hot research topics. One of the most important topic in automating the assembly tasks is the modeling of robotic assembly itself; that is adding the necessary skills to the robot that makes it aware of its surrounding environment using the wrench (Cartesian forces and torques) and pose (Cartesian position and orientation) signals of the manipulated object.

Contact States (CS) modeling of the force-controlled robotic tasks was solved by different approaches. Petri net was successfully employed in modeling and planning robotic assembly tasks and promising results were obtained [1]. In [2], fuzzy classifiers and neural networks were successfully employed in the recognition of different CS's using only the wrench signals. Modeling of peg-in-hole assembly process was successfully performed in the framework of finding analytical solutions of the contact forces for different situations between the manipulated object and the environment [3]. Hidden Markov models was successfully used in developing models for compliant motion robots and hence opening the door to the probabilistic modeling approaches [4,5]. In [6,8], the authors were successful in linking the CS modeling to the geometrical parameters estimation and efficient models were obtained for each CS. Stochastic Gradient Boosting (SGB) classifier was efficiently used in recognising different CS's without the need for knowing the task sequence or task graph [9]. In [11], the authors were successful in using fuzzy clustering technique in building efficient fuzzy models. The fuzzy clusters are tuned by Gravitational Search Algorithm (GSA) and excellent mapping capability was obtained for each model. A common feature to all of the approaches above is that the signals are considered stationary, i.e. their distribution is normal.

Selection and peer-review under responsibility of the International Scientific Committee of 5th CATS 2014 in the person of the Conference Chair Prof. Dr. Matthias Putz matthias.putz@iwu.fraunhofer.de
doi:10.1016/j.procir.2014.10.061

However, for the cases of robotic assembly in which the signals are non-stationary, then performance degradation would be resulted.

In this article, the Expectation Maximization-based Gaussian Mixtures Models (EM-GMM) [7] is used in modeling the force-controlled peg-in-hole assembly tasks. Through employing the EM-GMM, the non-normal distribution of the captured signals is accommodated through assigning multiple Gauss distributions for each signal. Furthermore, finding the parameters for each distribution is done through the EM algorithm that would increase the log-likelihood and improved performance would be obtained.

The rest of the paper is organized as follows; section 2 contains the description of the robotic peg-in-hole assembly process. Section 3 details the EM-GMM modeling process. Experimental validation on the assembly of camshaft caps is presented in section 4 and section 5 summarizes the concluding remarks and recommendation for future works.

2. Robotic Peg-in-Hole Assembly

Consider the robotic peg-in-hole assembly task shown in Fig. 1 that is composed of inserting a certain object (peg) into a certain hole and such a task is considered the backbone in many assembly tasks. In order to model the peg-in-hole task, the overall motion is segmented into different phases according to the location of the manipulated object with respect to the environment. For each segment different signals are collected and models are developed accordingly. Vision-based systems can be used in building the models for a robotic peg-in-hole assembly process. However, vision-based systems would fail for occluded parts and time varying illuminations that urged the researchers to consider developing the CS models using the wrench and pose signals that are measured by suitable sensors. Suppose that the wrench signals of the manipulated object, of the peg-in-hole assembly process shown in Fig. 1, are described as:

$$w=[f_x, f_y, f_z, \tau_x, \tau_y, \tau_z] \tag{1}$$

Where f_x, f_y, and f_z are the Cartesian forces and τ_x, τ_y, and τ_z are the torques around the Cartesian axes both measured for the manipulated object. Likewise to the pose of the manipulated object, it can be written as:

$$p=[x, y, z, \Psi_x, \Psi_y, \Psi_z] \tag{2}$$

with x, y, and z are the Cartesian position and Ψ_x, Ψ_y, and Ψ_z are the orientation around the Cartesian axes of the manipulated object. Hence, each classifier has 12 input signals, say $x_k=[x_{1,k},x_{2,k},\ldots,x_{12,k}]^T$. The CS classification problem can be formulated as:

$$y_k=\begin{cases}1 & if \quad x_k \in CS_k \\ 0 & Otherwise\end{cases} \tag{3}$$

y_k is the output of the k^{th} CS classifier. (3) is a nonlinear mapping between x_k and y_k and the goal of almost all modeling and classification researches is to approximate or realize this mapping as accurate as possible. In the next section, the proposed modeling approach, that approximates (3), will be explained.

3. Expectation Maximization-based Gaussian Mixture Models (EM-GMM)

Before detailing the EM-GMM process, the principles of the Bayesian modeling (or classification) is explained.

3.1. Bayesian Classification

Consider a vector set $x_k=[x_{1,k},x_{2,k},\ldots,x_{D,k}]^T$ with D to be the width of the vector. Suppose that the vector x_k belongs to one of the classes set $y_k=\{c_1,c_k,\ldots,c_C\}$. Then the vector x_k belongs to a class c_i implies that:

$$p(c_i \mid x_k) \geq p(c_j \mid x_k) \tag{4}$$

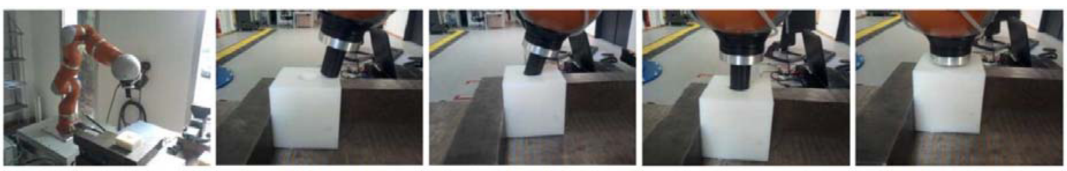
for $i\neq j$. $p(c_k|x_k)$ is called the a posterior probability of class c_k given the vector x_k and can be computed as:

$$p(c_i \mid x_k)=\frac{p(x_k \mid c_i)p(c_i)}{p(x_k)} \tag{5}$$

$p(x_k|c_i)$ is the probability density function (pdf) of class c_i in the vector space of x_k, $p(c_i)$ is the a priori probability that represents the probability of class c_i, and $p(x_k)$ is the probability of the vector space x_k which can be computed as:

$$p(x_k)=\sum_{i=1}^{C} p(x_k \mid c_i)p(c_i) \tag{6}$$

From (6), it can be seen that for equal class a priori $p(c_i)$, the term $p(x_k)$ of (5) would be merely a scaling factor. Therefore, one can say that the vector x_k belongs to a class c_i implies that



(a) (b) (c) (d) (e)

Fig. 1. Robotic peg-in-hole assembly phases: (a) phase 1(free space); (b) phase 2; (c) phase 3; (d) phase 4; (e) phase 5.

$$p(x_k \mid c_i)p(c_i) \geq p(x_k \mid c_j)p(c_j) \tag{7}$$

for $i \neq j$. According to (7), the best approximation of the term $p(x_k|c_i)$ results in the best classification for the pattern x_k. In the conventional Bayesian classifier, a Gauss distribution is used in approximating the term $p(x_k|c_i)$, that is:

$$p(x_k \mid c_i) = \frac{1}{2\pi^{D/2} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x_k - \mu)^T \Sigma^{-1}(x_k - \mu)) \tag{8}$$

where $\mu \in R^D$ is the mean, $\Sigma \in R^{D*D}$ is the covariance matrix, and $|\Sigma|$ is the determinant of Σ. It was shown that the approximation (8) performs well for the case of normal distribution. However in many cases, the signals of the given vector space, or several signals of the vector space, have non-normal distribution and consequently the use of (8) would result in increased misclassifications.

3.2. Gaussian Mixtures Model (GMM)

In order to accommodate the possible non-normal distribution of the signals, mixtures of Gaussian components are used in modeling the features, i.e. assigning multiple Gauss distributions for each feature. Suppose that a single Gaussian distribution is represented as:

$$N(x_k, \mu, \Sigma) = \frac{1}{2\pi^{D/2} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(x_k - \mu)^T \Sigma^{-1}(x_k - \mu)) \tag{9}$$

Then a Gaussian Mixtures Model (GMM) can be described as:

$$p(x_k \mid c_i) = \sum_{q=1}^{M} \omega_q N_q(x_k, \mu_q, \Sigma_q) \tag{10}$$

M is the total number of Gaussian mixtures; ω_q, μ_q, and Σ_q are the weight, mean, and covariance of the q^{th} Gaussian component. Suppose that $\theta_q = (\omega_q, \mu_q, \Sigma_q)$ and consider the parameter vector $\theta = [\theta_1, \theta_2, \ldots, \theta_M]^T$. One can see that finding the values of the parameters is very important in having a precise modeling of the given features. Therefore, the models (10) can be written in terms of the parameters θ as:

$$p(x_k \mid c_i; \theta) = \sum_{q=1}^{M} \omega_q N_q(x_k, \mu_q, \Sigma_q) \tag{11}$$

Hence, finding the parameter vector θ that optimizes the models from the available measurements would optimize the performance of the classification process.

3.3. Expectation Maximization (EM)

One of the most efficient approaches in finding those parameters is the Expectation Maximization (EM) algorithm. The EM algorithm is composed of two steps; the E-step in which the log-likelihood is estimated using the current parameters, and the M-step that updates the parameters θ such that the log-likelihood is maximized. In order to explain the EM algorithm, let's consider the overall data $X = [x_1, x_2, \ldots, x_N]^T$, then the log-likelihood can be computed as [7]:

$$L(X \mid c_i; \theta) = \sum_{n=1}^{N} \ln(p(x_n \mid c_i; \theta)) \tag{12}$$

The parameters θ that maximizes (12) can be described as:

$$\theta = \arg(\max_{\theta} \{L(X \mid c_i; \theta)\}) \tag{13}$$

Analytical solution for (13) are intractable, therefore (13) is normally solved iteratively and the EM algorithm is considered one of the prominent approaches in computing the values of θ that achieves (13). Below is a summary of the EM algorithm:

Step 1: Initialize the parameter vector $\theta_q = (\omega_q, \mu_q, \Sigma_q)$. Initialize the convergence parameters ε and δ.

Step 2: (E-Step) Use the current parameter vector θ_q and compute the responsibilities that are:

$$\gamma_n = \frac{\omega_q N_q(x_n, \mu_q, \Sigma_q)}{\sum_{q=1}^{M} \omega_q N_q(x_n, \mu_q, \Sigma_q)} \tag{14}$$

Step 3: (M-Step) Re-estimate the parameters using the current responsibilities:

$$\mu_q^{new} = \frac{1}{N_q} \sum_{n=1}^{N} \gamma_n x_n \tag{15}$$

$$\Sigma_q^{new} = \frac{1}{N_q} \sum_{n=1}^{N} \gamma_n (x_n - \mu_q^{new})(x_n - \mu_q^{new})^T \tag{16}$$

$$\omega_q^{new} = \frac{N_q}{N} \tag{17}$$

with:

$$N_q = \sum_{n=1}^{N} \gamma_n \tag{18}$$

Step 4: Compute the log-likelihood:

$$\ln p(X;\theta) = \sum_{n=1}^{N} \ln\{\sum_{q=1}^{M} \omega_q N_q(x_n, \theta)\} \tag{19}$$

Step 5: Check for the convergence:
If $|\theta^{new}-\theta| \leq \varepsilon$ or $|ln\ p(X;\theta^{new})-ln\ p(X;\theta)| \leq \delta$ then stop.
Otherwise go to **Step 2**.

For more details on the EM-GMM and the derivation of the equations above, see [7]: chapter 9. In the next section, the performance of the EM-GMM based classifier will be evaluated for the CS recognition of the cylinder head camshaft caps assembly of a powertrain.

4. Experimental Results

For validating the suggested modelling strategy, a test stand is built that is composed of a KUKA Lightweight Robot (LWR) 4+ performing the camshaft caps assembly process of a cylinder head of an automotive powertrain. The key features of the KUKA LWR 4+ are detailed in [10]. For research purposes a Fast Research Interface (FRI) port is available in the robot that enables researchers of measuring the wrench and pose of the manipulated object by sensors installed within the robot arm. The FRI port is connected to a remote PC that performs the computational aspects of the modeling process. The features of the PC that was used is of Intel (R) Core (TM) i5-2540 CPU with 2.6 GHz speed and 4 GB RAM running under a Linux environment. The rate of the communication between the remote PC and the robot, through the FRI, is 100 Hz. The programming is done through a C++ platform. Fig. 2 shows five distinct phases of a peg-in-hole robotic assembly task and the aim is to use the EM-GMM approach in modeling those phases using the wrench and pose measurements of the manipulated object. Fig. 3 shows the captured wrench and pose signals when performing the task depicted in Fig. 2. The number of samples of phases 1, 2, 3, 4, and 5 are 1092, 2332, 4077, 2212, and 992 samples respectively. For each phase, 250 samples were taken out for test and the remaining samples were used in building the models. That is for phases 1, 2, 3, 4, and 5 the training samples were 842, 2082, 3827, 1962, and 742 samples respectively.

Increasing the number of mixtures would improve the modeling accuracy. However, the computational time would be increased. Hence, 3 mixtures were chosen in order to have a good EM-GMM modeling accuracy with reasonable computational time. In order to perform comparison with other approaches, the same modeling process was carried out using the Conventional Fuzzy Classifier (CFC) [2], the Stochastic Gradient Boosting Classifier (SGB) [9] and the Gravitational Search- Fuzzy Clustering Algorithm (GS- FCA) [11]. Fig. 4 shows the models outputs when using the EM-GMM. In Fig. 5, the corresponding models outputs were plotted when using the CFC, SGB, and GS-GCA schemes. The Classification Success Rate (CSR) was computed when using the test samples for all approaches above and it was found to be 97.4%, 73.6%, 68.7%, and 32.5% for the EM-GMM, GS-FCA, SGB, and CFC recognition schemes. It is clear that the EM-GMM modeling scheme is outperforming the rest. To grasp the real reason behind the excellent performance of the EM-GMM based modeling, phase 5 is considered as a sample and plotted the histogram of all signals for this phase of the peg-in-hole assembly task as shown in Fig. 6. One can see that almost all signals are non-stationary, i.e. they are abnormally distributed that gives the privilege to the EM-GMM in modeling such signals. Furthermore, the use of the EM algorithm in finding the models parameters would make the approach more accurate since the log-likelihood is maximized as explained in section 2. Both those two factors add complementing features to the EM-GMM scheme in modeling such a process. It is worth noting that in the EM algorithm, the initialization of the parameter vector θ is done randomly. The initialization could have an influence on the modeling accuracy. However, such influence is not so significant that could degrade the modeling performance. That is to say if the initial estimate greatly affects the final result, then a significant performance degradation would be observed at least for one of the phases (which was not the case). Furthermore, the computational time of building the models was measured for the CFC, EM-GMM, SGB, and GS-FCA to be 0.0014, 26.635, 129.899, and 333.184 sec respectively. It can be noticed that the CFC modeling scheme has the least computational time with a degraded Classification Success Rate (CSR) of 32.5% that makes it undesirable. Compared with the SGB and the GS-FCA modeling schemes the EM-GMM is of reduced computational time with enhanced CSR performance. Table -1- summarizes the CSR and the computational time of all approaches considered in this article.

Table 1. Classification Success Rate (CSR) and the Computational Time for the EM-GMM, GS-FCA, SGB, and CFC approaches.

Approach	Classification Success Rate (CSR) (%)	Computational Time (Sec)
EM-GMM	97.4	26.635
GS-FCA	73.6	333.184
SGB	68.7	129.899
CFC	32.5	0.0014

(a) (b) (c) (d) (e)

Fig. 2. Camshaft caps assembly phases: (a) phase 1(free space); (b) phase 2; (c) phase 3; (d) phase 4; (e) phase 5.

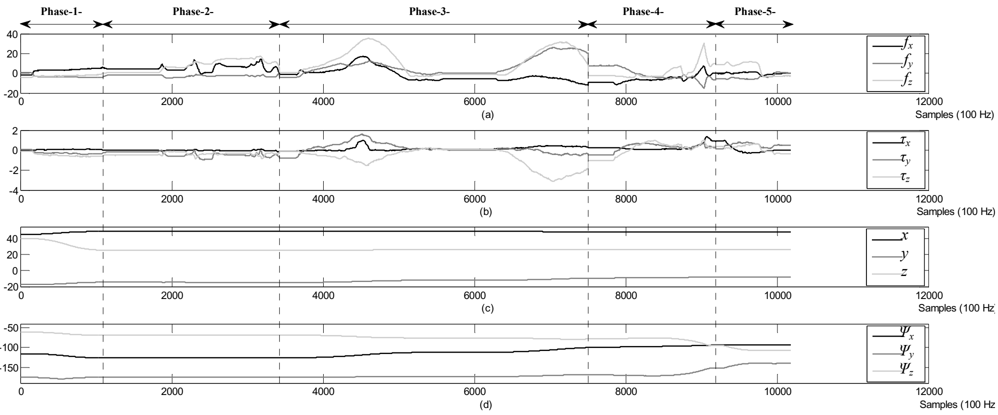


Fig. 3. The manipulated object wrench and pose measurements: (a) Cartesian forces (in N); (b) Torques around the Cartesian axes (in N.m); (c) Cartesian position (in cm); (d) Orientation around the Cartesian axes (in degree).

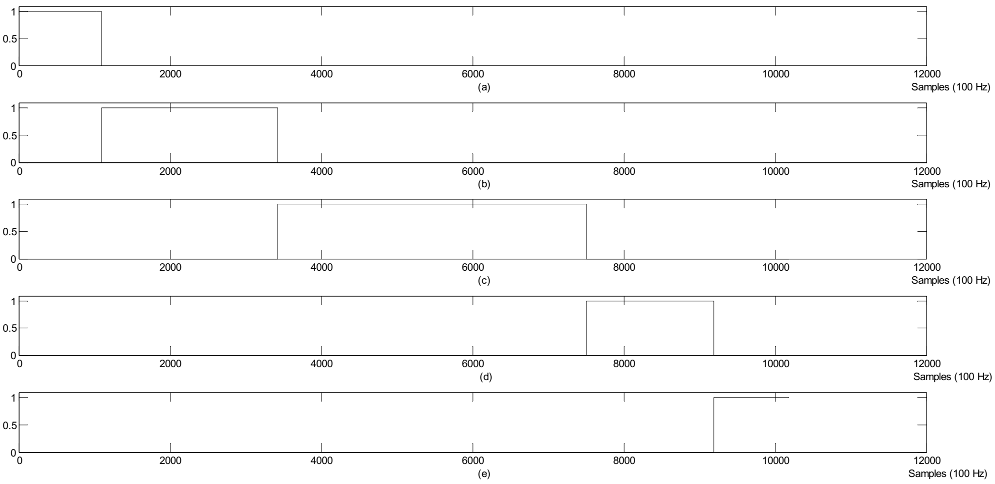


Fig. 4. EM-GMM based models outputs: (a) phase 1(free space); (b) phase 2 (c) phase 3; (d) phase 4; (e) phase 5.

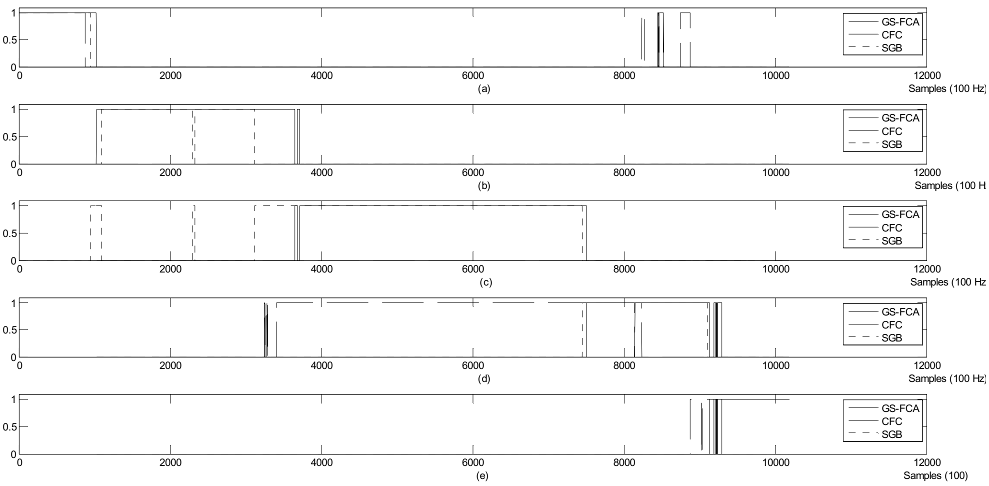


Fig. 5. GS-FCA (solid), SGB (dashed), and CFC models outputs (hidden): (a) phase 1(free space); (b) phase 2 (c) phase 3; (d) phase 4; (e) phase 5.

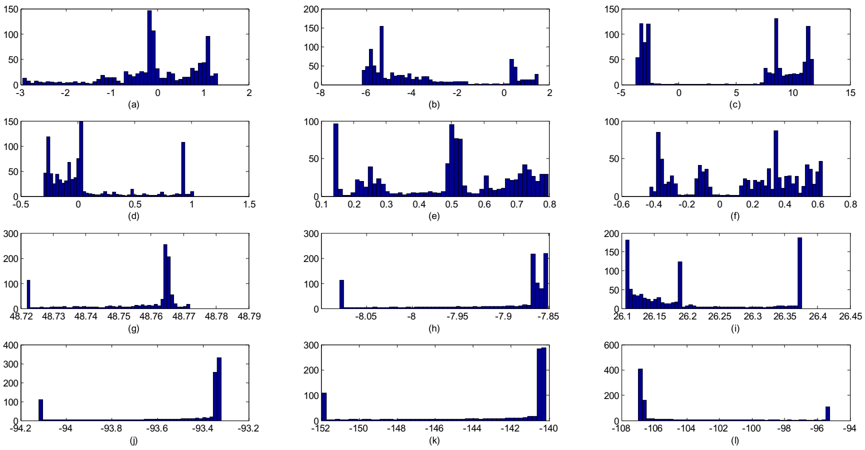


Fig. 6. Histogram of phase 5: (a) f_x; (b) f_y; (c) f_z; (d) τ_x; (e) τ_y; (f) τ_z; (g) x; (h) y; (i) z; (j) Ψ_x; (k) Ψ_y; (l) Ψ_z.

5. Conclusion

The Expectation Maximization-based Gaussian Mixtures Model (EM-GMM) was successfully employed in modeling the force-controlled robotic assembly tasks. The robotic assembly task of camshaft caps was segmented into five phases and for each phase the wrench (Cartesian forces and torques) and pose (Cartesian position and orientation) of the manipulated object was collected. Using the EM-GMM approach, excellent models were obtained. A KUKA Lightweight Robot (LWR) 4+ was used in the experimental validation. Comparison with the available assembly modeling approaches, like Gravitational Search-Fuzzy Clustering Algorithm (GS-FCA), Stochastic Gradient Boosting (SGB), and Classical Fuzzy Classifier, was also considered. It was shown that the EM-GMM is outperforming the rest. The superiority of the EM-GMM originates from considering the non-normality in the signals distribution and the use of the EM algorithm in finding the models parameters. Furthermore, it was shown that the EM-GMM modeling scheme is of reduced computational time compared with the SGB and GS-FCA modeling schemes. Possible improvement on the EM-GMM modeling is to integrate the feature selection technique within this modeling approach that would enhance the accuracy and make it more robust against possible variations. However, such a study is left to future works.

Acknowledgements

This work is supported by the Fonds National de la Recherche (FNR) in Luxembourg under grant no. AFR-2955286.

References

[1] McCarragher B.J., 1994. Petri net modeling for robotic assembly and trajectory planning. IEEE Trans. Indust. Elect., vol. 41, no. 6, p. 631- 640.

[2] Skubic M., Volz R.A., 2000. Identifying single-ended contact formations from force sensor patterns. IEEE Trans. Robot. Autom., vol. 16, no. 5, p. 597-603.

[3] Fei Y., Zhao X., 2003. An assembly process modeling and analysis for robotic multiple peg- in- hole. J. Intel. Robot. Syst., vol. 36, p. 175-189.

[4] Lau H.Y.K., 2003. A hidden Markov model- based assembly contact recognition system. Mechatronics, vol. 13, 2003, p. 1001-1023.

[5] Debus T.J., Dupot P.E., Howe R.D., 2004. Contact state estimation using multiple model estimation and hidden Markov models. Int. J. Robot. Res., vol. 23, no. 4, p. 399-413.

[6] De Schutter J., Bruyninckx H., Dutré S., De Geeter J., Katupitiya J., Demey S., Lefenvre T., 1999. Estimating first order geometric parameters and monitoring contact transitions during force controlled compliant motion. Int. J. Robot. Res., vol. 18, no. 12, p. 1161-1184.

[7] Bishop C.M., 2006. Pattern Recognition and Machine Intelligence. Springer.

[8] Meeussen W., Rutgeerts J., Gadeyne K., Bruyninckx H., De Schutter J., 2007. Contact- state segmentation using particle filters for programming by human demonstration in compliant- motion tasks. IEEE Trans. Robot., vol. 23, no. 2, p. 218-231.

[9] Cabras S., Castellanos M.E., Staffetti E., 2010. Contact- state classification in human- demonstrated robot compliant motion tasks using the boosting algorithm. IEEE Trans. Syst. Man, Cybern. B, vol. 40, no. 5, p. 1372-1386.

[10] Lightweight Robot 4+ Specification, Version: Spez LBR 4+ V5 en, KUKA Roboter GmbH, 22 Dec., 2011.

[11] Jasim I.F., Plapper P.W., 2013. T-S fuzzy contact state recognition for compliant motion robotic tasks using gravitational search-based clustering algorithm. IEEE Int. Conf. Fuzzy Systems, Hyderabad- India, 7-10 July.