T-S Fuzzy Contact State Recognition for Compliant Motion Robotic Tasks Using Gravitational Search-Based Clustering Algorithm

Ibrahim F. Jasim
Faculty of Science, Technology, and Communication
University of Luxembourg
L- 1359 Luxembourg
E-mail: ibrahim.jasim@uni.lu

Peter W. Plapper
Faculty of Science, Technology, and Communication
University of Luxembourg
L-1359 Luxembourg
E-mail: peter.plapper@uni.lu

Abstract— **In this paper, we address the problem of contact state recognition for compliant motion robotic systems. The wrench (Cartesian forces and torques) and pose (position and orientation) of the manipulated object in different Contact Formations (CFs) are firstly captured during a certain task execution. Then for each CF, we develop an efficient Takagi-Sugeno (T-S) fuzzy inference system that can model that specific CF using the available input (wrench and pose) - output (the desired model output for each CF) data. The antecedent part parameters are computed using the Gravitational Search- based Fuzzy Clustering Algorithm (GS- FCA) and the consequent parts parameters are tuned by the Least Mean Square (LMS). Excellent mapping and hence recognition capabilities can be expected from the suggested scheme. In order to validate the approach; experimental test stand is built which is composed of a KUKA Light Weight Robot (LWR) manipulating a cube rigid object that interacts with an environment composed of three orthogonal planes. The manipulated object is rigidly attached to the robot arm. The robot is programmed, by a human operator, to move in different CFs and for each CF, the wrench and pose readings are captured via the Fast Research Interface (FRI) available at the KUKA LWR. Using the suggested approach, excellent modeling is obtained for different CFs during the robot task execution. A comparison with the available CF recognition approaches is also performed and the superiority of the suggested scheme is shown.**

Keywords—compliant motion robots; contact state; fuzzy clustering; gravitational search

I. INTRODUCTION

Compliant motion robots are very crucial in many applications like assembly, blurring, space…etc. In a compliant robotic system, the object manipulated by the robot is subject to contacts with its surrounding environment. In [1], Desai and Volz proposed a significant milestone in the advent of the compliant motion robotic systems through introducing the notion of the Contact Formation (CF). For instance, if we have a polyhedral manipulated object interacting with a certain environment, then we can describe the contact of the object vertex to the face of the environment and call it as a vertex-face *(v-f)* contact. Similarly, for edge- face (*e-f*), face- face (*f-f*), 2 faces-edge-face (*ff-ef*), 2 faces- 2faces (*2f-2f*), 3 faces- 3faces (*3f-3f*), and other possible contacts. Each one of those contacts phases is called a Contact Formation (CF), and a compliant motion robotic task can be composed of a sequence of CFs.

Depending on wrench (Cartesian forces and torques), twists (linear and angular velocities), and/or pose (position and orientation) of the manipulated object, researchers in the field of robotics and automation were capable of analyzing those quantities and mapping them into their corresponding CFs. Different recognition approaches were used like Petri net [2, 3], Hidden Markov Models (HMM) [5-7], Bayesian and particle filters [8, 9], Auto Regressor eXogenous (ARX) modeling [10], and Stochastic Gradient Boosting (SGB) [11], particle filters and wrench space computation from the CF graph [12], and fuzzy classifiers [13- 16].

Despite their excellent performance, the approaches above are having drawbacks in several senses. For instance, the Petri net proved to be good in recognition and planning, but its implementation is complex and the task sequence is required to be known. For the probabilistic approaches, they are of simpler implementation but the task sequence is required to be known. Regarding the Stochastic Gradient Boosting (SGB) approach, it relaxed the need for the task sequence, but it relies on minimizing a non-convex function that may cause trapping into local minima and neglecting several signals that are considered as unimportant that would make the classifier to depend on less signals and hence increasing input signals space overlap from different CFs. For fuzzy approaches reported in [13- 16], they contain one if-then rule for each class which would constitute a critical incapability for the fuzzy classifier to map the input-output data. Furthermore, the estimation of the membership functions parameters relied only on computing the mean and the standard deviation of the available measurements for each signal. In [17], Breiman suggested that injecting the randomness for functions approximation can highly improve the performance of the approximation processes. So, integrating the fuzzy classifiers suggested in [13- 16] with randomized algorithms, in finding the best mapping, can highly enhance the fuzzy classification process. Moreover, as the number of CFs is increased, a more efficient mapping capability is required which makes the use of a single rule for each CF to be incapable of well recognizing different CFs.

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

In this paper, we develop a new CF recognition system that relies on building a T-S fuzzy model with multiple *if- then* rules for each CF. The antecedent part parameters for each model are computed by the Gravitational Search- Fuzzy Clustering Algorithm GS-FCA approach [18, 19]. The LMS is used in tuning the consequent part parameters for each *if-then* rule of each CF. The main contribution of this paper is to have a CF recognition system with the following features:

1. The suggested approach doesn't require knowing the CFs sequence or graph.
2. Compared to the approaches in [13- 16], enhanced input-output mapping through using:
   i. GS- FCA and LMS in tuning the T-S fuzzy models.
   ii. Multiple rules are used for each CF model.

The rest of the paper is organized as follows. In section 2, we describe the T-S fuzzy classifier. Section 3 details the computations of the T-S fuzzy classifier parameters. Experimental results are given in section 4. Conclusion and recommendations for future works are stated in section 5.

II. T-S Fuzzy Classifier

Suppose that it is required to identify an unknown nonlinear system:

$$y = f(\mathbf{x}) \tag{1}$$

that maps some available input- output data, say $\mathbf{x}_k = [x_{1,k}, \ldots, x_{n,k}]$ and y_k with k $(k = 1,2, \ldots \bar{N})$ is the index of the training pairs. The nonlinear system (1) can be efficiently modeled using a T-S fuzzy system that breaks the nonlinear system into sum of linear models each one of them is described by the following *if-then* rules [20]:

$$R_i\text{: If } x_1 \text{ is } A_{i,1}(x_1) \text{ and } \ldots \text{ and } \ x_n \text{ is } A_{i,n}(x_n)$$
$$\text{then } \hat{y} = \boldsymbol{a}_i^T\mathbf{x} + b_i \tag{2}$$

$A_{i,j}(x_j)$ is a membership function that quantifies for the input x_j. $\boldsymbol{a}_i$ and b_i are the parameters of the i$^{\text{th}}$ local linear model. (2) can be aggregated using the fuzzy mean approach and the output can be computed as:

$$\hat{y} = \frac{\sum_{i=1}^{c}\beta_i(\mathbf{x})(\boldsymbol{a}_i^T\mathbf{x}+b_i)}{\sum_{i=1}^{c}\beta_i(\mathbf{x})} \tag{3}$$

Where:

$$\beta_i(\mathbf{x}) = \prod_{j=1}^{n} A_{i,j}(x_j) \tag{4}$$

c is the number of rules for each model. If we use Gauss membership functions for the antecedent part of (2), then we have:

$$A_{i,j}(x_j) = \exp\left(-\frac{(x_j-\delta_{ij})^2}{2\sigma_{ij}^2}\right) \tag{5}$$

δ_{ij} and σ_{ij}^2 are the membership function center and variance respectively. For the robotic system under consideration, it is required to classify the CFs according to the available wrench and pose readings. The inputs to the classifier are the wrench vector:

$$\boldsymbol{w} = [f_x, f_y, f_z, T_x, T_y, T_z] \tag{6}$$

and pose vector:

$$\boldsymbol{p}_o = [x, y, z, \Psi_x, \Psi_y, \Psi_z] \tag{7}$$

f_x, f_y and f_z are the Cartesian forces, T_x, T_y and T_z are the torques around the Cartesian axes, x, y and z represent the Cartesian position, and Ψ_x, Ψ_y, and Ψ_z are the orientation of the manipulated object. So, we have 12 input signals for each classifier, say $\mathbf{x}_k = [x_{1,k}, \ldots, x_{12,k}]$. Now, the CF classification problem can be formulated as:

$$y_k = \begin{cases} 1 & \text{if} \quad \mathbf{x}_k \in \text{current CF} \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

Equation (8) represents a nonlinear mapping of the form $y_k = f(\mathbf{x}_k)$ explained in (1)- (5), and throughout this paper; we will focus on estimating $f(\mathbf{x}_k)$ using the T-S fuzzy model. However, the approximation accuracy would highly rely on the choice of the antecedent and consequent parts parameters for each *if-then* rule. In the following section, we will describe how we can choose those parameters for an optimal classification process.

III. T-S Fuzzy Classifier Parameters Computation

A. Antecedent Part Parameters:

The GS- FCA will be used in computing the antecedent part parameters, say the membership functions parameters. Suppose that we have a data set $\boldsymbol{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_L}\} \subset R^p, L, p \in N$. Then, with clustering, the data set $\boldsymbol{X}$ can be grouped into clusters in which the data group of each cluster share a certain attribute, i.e. grouping the data set $\boldsymbol{X}$ into $c \in \{2, \ldots, L-1\}$ clusters. Fuzzy clustering performs the function of data clustering using the concept of fuzzy sets theory. Each element of the set $\boldsymbol{X}$ is assigned with a membership function that quantifies the degree of its affiliation to one of the given clusters. Consider that μ_{ij} represents the membership function that quantifies strength in which the i$^{\text{th}}$ data set of $\boldsymbol{X}$ belongs to the j$^{\text{th}}$ cluster, then $\mu_{ij} \in [0,1], (i = 1,2, \ldots, n, j = 1,2, \ldots, c)$. Extending μ_{ij} for all data sets and over all clusters, a partition matrix, denoted as $U \in M$, is obtained which can be described by:

$$M = \{U: U \in [0,1]^{n\times c}\} \tag{9}$$

However, the following constraints are necessary to be satisfied for the selection of the partition matrix.

$$\sum_{j=1}^{c}\mu_{ij} = 1, \ i = 1,2, \ldots, n \tag{10}$$

$$\sum_{i=1}^{n}\mu_{ij} > 0, \ j = 1,2, \ldots, c \tag{11}$$

The fuzzy clustering problem can be solved by finding the clusters centers and the partition matrix. Fuzzy c-means (FCM) clustering is a widely used approach in which both the clusters centers and the partition matrix are found through solving the following constrained optimization problem:

$$minimize\ J_f = \sum_{j=1}^{c}\sum_{i=1}^{n}\mu_{ij}^{m}\left\|v_j - x_i\right\|_2 \quad (12)$$

s.t. (10) and (11).

with $m > 1$, v_j is the j[th] cluster center, and $\|.\|_2$ represents a norm on R, frequently Euclidean norm is utilized. It is worth noting that x_i could be a vector of signals (which is the case of our application) and clustering is then performed in a vector wise for the clusters centers, that is each cluster center would be a vector of dimension equal to the number of columns of the vector x_i. Alternating Optimization (AO) was successfully used to solve the constrained optimization above and the following solution was obtained [21]:

$$\mu_{ik} = \frac{1}{\sum_{j=1}^{c}\left(\frac{\|v_k - x_i\|_2}{\|v_j - x_i\|_2}\right)^{\frac{2}{m-1}}} \quad (13)$$

and

$$v_j = \frac{\sum_{i=1}^{n}\mu_{ij}{}^{m}x_i}{\sum_{i=1}^{n}\mu_{ij}{}^{m}} \quad (14)$$

However, it was found that such a solution can be easily trapped in local minima because they depend on derivatives in their optimization process [22]. Using a more powerful stochastic optimization algorithm can highly improve the fuzzy clustering algorithm above. Gravitational Search Algorithm (GSA) is a recently developed efficient stochastic optimization algorithm that depends on the concept of Newton laws of gravity and motion. It is assumed that all agents within a certain population have certain gravitational forces between them, and the agent with higher mass exerts higher force and becomes more optimal (see [23] for more details about the GSA and its performance). Suppose that we have N agents and the p[th] agent is described by:

$$\boldsymbol{z}_p = \left(z_p^1, \dots, z_p^m\right) \quad (15)$$

$$p = 1,2, \dots N$$

z_p^k is the position of the p[th] agent in the k[th] dimension. Let's define the gravitational force between the p[th] and the q[th] agents to be:

$$F_{pq}^d(t) = G(t)\frac{M_p(t)\times M_q(t)}{\|\boldsymbol{z}_p, \boldsymbol{z}_q\|_2}\left(z_q^d - z_p^d\right) \quad (16)$$

Where $M_p(t)$ and $M_q(t)$ are the masses, z_p^d and z_q^d are the positions of the p[th] and q[th] agents respectively, and $G(t)$ is the gravitational constant. The gravitational constant can be described by the following expression:

$$G(t) = G_o.exp\left(-\alpha\frac{t}{max\_t}\right) \quad (17)$$

G_o is the initial value of the gravitational constant, α is a constant, t is the current iteration, and *max_t* is the maximum iteration. The inertial mass for the p[th] agent can be computed as:

$$M_p(t) = \frac{m_p(t)}{\sum_{q=1}^{N}m_q(t)} \quad (18)$$

m_p can be found as:

$$m_p(t) = \frac{fit_p - worst}{best - worst} \quad (19)$$

fit_p is the value of the fitness function (objective function) for the p[th] agent, *best* and *worst* have different expressions depending on the nature of the optimization problem in hand, i.e. minimization or maximization optimization problems. For minimization problems *best=min* (fit_p) and *worst=max* (fit_p), and for maximization problems *best=max* (fit_p) and *worst=min* (fit_p). The force exerted on the p[th] agent can be computed as a random weighted sum of all attraction forces from other agents, i.e.

$$F_p^d(t) = \sum_{p\neq q} rand_q.F_{pq}^d(t) \quad (20)$$

$rand_q$ is a random number. Using the Newton law of motion, we can find the acceleration of the p[th] agent movement as:

$$a_p^d(t) = \frac{F_p^d(t)}{M_p(t)} \quad (21)$$

The new velocities and positions can be computed according to the following equations:

$$v_p^d(t+1) = rand_p.v_p^d(t) + a_p^d(t) \quad (22)$$

$$z_p^d(t+1) = z_p^d(t) + v_p^d(t+1) \quad (23)$$

The GSA optimization above has proved to be efficient in solving the clustering problems [18, 19]. Furthermore, GSA is a random search method that doesn't depend on the computations of derivatives and consequently the problem of trapping in local minima can be avoided. Moreover, GSA can be used to solve optimization problems in which the objective function is non-differentiable, and this opens the door to its applicability to discontinuous functions identification. However, the dimension of the data to be clustered may restrict the usability of the GSA algorithm for finding only the clusters centers since using the GSA in finding the partition matrix would be tedious for high dimensional data. Therefore, in order to minimize the objective function given in (12), we will use (13) for computing the partition matrix, but for clusters centers, instead of using (14), we will utilize the GSA in minimizing the objective function (12). The algorithm below details the GS- FCA through which the centers of clusters are computed using the GSA optimization:

Algorithm 1 (GS-FSA):

Step 1: Set l=1, initialize the centers and code them into positions of agents. Initialize the tolerance ε. Initialize U^l.

Step 2: Compute the objective function for each agent using (12). Update l as l=l+1.

Step 3: Update G using (17) and find the *best* and *worst* of the agents.

Step 4: Compute the mass M_p and then calculate the gravitational force for each agent F_p using (20).

Step 5: Compute the acceleration for each agent using (21).

Step 6: Update the velocity and position for each agent according to (22) and (23) respectively.

Step7: Compute the partition matrix $U^l = [\mu_{ij}]^{n\times c}$ using (13).

Step 8: If $|U^l - U^{l-1}| < \varepsilon$ then stop. Otherwise repeat **Steps 2-8**.

As per accomplishing Algorithm 1, the center and variance for each membership function can be computed as [24]:

$$\delta_{ij} = \frac{\sum_{k=1}^{L} \mu_{kj} \cdot x_{kj}}{\sum_{k=1}^{L} \mu_{kj}} \tag{24}$$

$$\sigma_{ij} = \sqrt{\frac{\sum_{k=1}^{L} \mu_{kj} \cdot (x_{kj} - \delta_{ij})}{\sum_{k=1}^{L} \mu_{kj}}} \tag{25}$$

B. Consequent Parts Parameters:

The consequent parts parameters, say $\boldsymbol{a}_i$ and b_i, are tuned using the Least Mean Square (LMS) algorithm. Suppose that $\boldsymbol{\theta}_i = [\boldsymbol{a}_i \; b_i]$, then the value of $\boldsymbol{\theta}_i$ can be computed as :

$$\boldsymbol{\theta}_i^* = \underset{\boldsymbol{\theta}_i}{\arg\min} \; \frac{1}{N}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}_i)^T \boldsymbol{\Phi}_i (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\theta}_i) \tag{26}$$

Where $\boldsymbol{X} = [\mathbf{x}\;\mathbf{1}]$ and $\boldsymbol{\Phi}_i$ is a diagonal matrix with the membership grades are the elements of the main diagonal [25]:

$$\boldsymbol{\Phi}_i = \begin{bmatrix} \mu_{i1} & 0 & \cdots & 0 \\ 0 & \mu_{i2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \mu_{iL} \end{bmatrix} \tag{27}$$

$\boldsymbol{\Phi}_i$ is obtained from Algorithm 1.

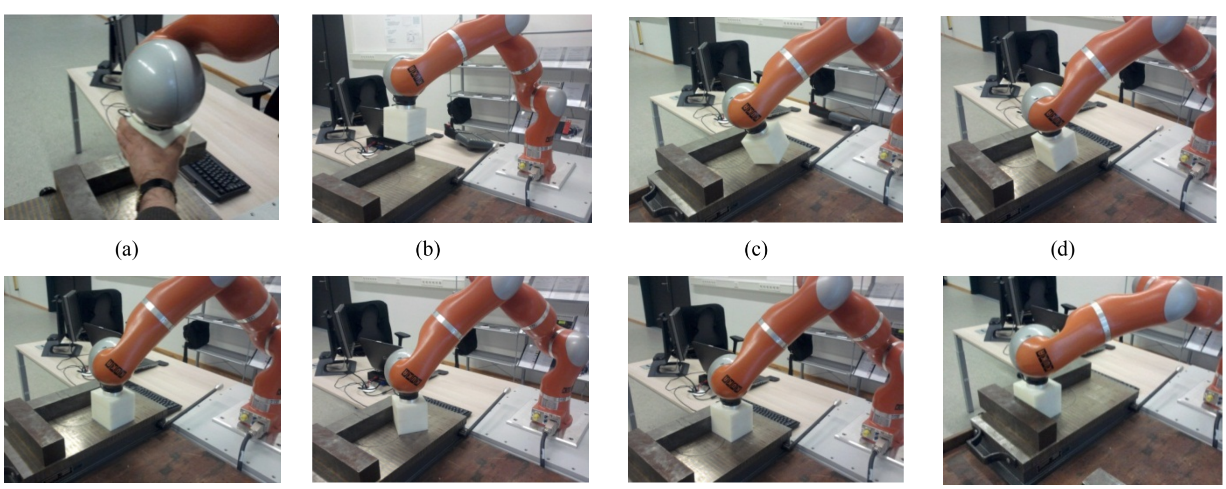
Then using the LMS, the parameters can be updated as:

$$\boldsymbol{\theta}_i = (\boldsymbol{X}^T \boldsymbol{\Phi}_i \boldsymbol{X})^{-1} \boldsymbol{\Phi}_i \boldsymbol{X}^T \boldsymbol{\Phi}_i \boldsymbol{y} \tag{28}$$

Through this LMS training, the models are tuned to be fitted more to their desired input- output mappings. This will be clearly noticed from the performance of the suggested approach throughout the experiments below.

IV. Experimental Results

The test stand of the experiment is composed of a KUKA Light Weight Robot (LWR) manipulating a cube rigid object. The KUKA LWR is programmed to move the object in different positions such that different CFs are realized with respect to an environment composed of three orthogonal planes. Fig. 1 shows the test stand used in this experiment. During a robot task execution, different CFs could be brought about. Fig. 1 (a) shows a human operator guiding the robot for adding recognition skills to the robot. Fig. 1 (b) to (h) show seven of the possible CFs (other CFs could be dealt with in a similar manner). The wrench and pose readings of the manipulated object are captured through the Fast Research Interface (FRI), available in the KUKA LWR robot, with a sampling rate of 100 Hz. In order to model each CF successfully, an output, say y_h, is assigned for the h^{th} CF which needs to be equal to 1 if the manipulated object is currently in that specific CF and 0 otherwise. The desired output y_h and the captured wrench and pose signals for each CF are collected. For the sake of showing the performance of the suggested approach in recognizing different CFs, the manipulated object was moved in the sequence (b-c-b-d-b-e-f-g-h-b) of Fig. 1. As mentioned previously, the CFs sequence is not required in building the CFs model through the suggested approach, but we followed this sequence just for demonstration and we will show later that the developed models can be efficiently used to any other CFs sequence. So, we have 7 CFs to be recognized, say *free space*, *v-f*, *e-f*, *f-f*, *2f-ef*, *2f-2f*, and *3f-3f*. In the training phase, we segmented the overall task into segments each segment correspond to a CF. Totally we had 10 segments (4 belong to the *free space* CF and the other 6 belong to the rest of the considered CFs). Using the suggested GS-FCA based TS fuzzy classifier, a model is built for each CF from the available input-output data. The number of clusters of the input signals was considered to be seven, and the number of the if-then rules for each CF was considered to be equal to the number of clusters. If we increase, the number of clusters (and hence the number of rules for each CF), then we can expect to obtain a more



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

(e) (f) (g) (h)

Fig. 1. (a) skills are added by human guidance. (b) *free space* CF. (c) *vertex- face (v-f)* CF. (d) *edge- face (e-f)* CF. (e) *face- face (f-f)* CF. (f) *2face-edge-face (2f-ef)* CF. (g) *2face-2face (2f-2f)* CF. (h) *3face- 3face (3f-3f)* CF.

enhanced mapping capability of the models, however, the computation time would be also increased. In the test phase, we repeated the experiment mentioned above and the wrench and pose readings captured during the test phase are shown in Fig. 2. Note that the signals we got in training and test are having high similarity and due to space limitation we inserted only the test signals. Using the signals of Fig. 2 as inputs to each CF model developed in the training phase, the output of all models were computed. Furthermore, for comparison purposes, we built the corresponding CFs models using the conventional Fuzzy Classifier (FC) suggested in [13- 16], and SGB Classifier (SGBC) suggested in [11]. The target y_h, the models outputs for the GS-FCA TS, SGB, and the conventional fuzzy classifiers for all the CFs of this task are shown in Fig. 3. It is clear that the suggested TS fuzzy classifier is of superior performance if compared to the available CF recognition schemes. Examining Fig. 3 can assist in deducing the following comments:

C1. For the recognition approach suggested in this paper, a model is derived for each one of the CFs that is working independently of other CFs models. Consequently, the CFs sequence is not required to be known for the robotic task. Conversely for the approach depicted in [8], the CF sequence is required to be known that adds more limitation to the CF recognition system (especially for applications that have possible variable sequence of CFs execution). Furthermore, we performed another task that have the sequence of contacts (e)-(g)- (e)- (b) shown in Fig. 1, i.e. the CFs would be in the sequence *(f-f)-(2f-2f)-(f-f)-free space*. The wrench and pose signals captured during this task are shown in Fig. 4. We used the same models developed in the previous task and the results are shown in Fig. 5. It is obvious that even though the sequence has a significant change from the previous one, the suggested CFs models results in excellent performance. Of course, the approach suggested in [8] is completely inapplicable since the recognition process relies on the task sequence.

C2. In [9], even though it is not required to know the exact sequence of the CFs within a certain task, the CFs should follow a prescribed sequence, otherwise it will cause a significant impact on the number of particles of the filters causing a performance limitation. Whereas in the GS- FCA TS classifier approach, it is not necessary for the CFs to follow a prescribed sequence since each model can efficiently produce 1 or 0 depending on the current CF and without regard to the sequence or previous CFs.

C3. When using the SGB classifier for recognizing the CFs of a robotic system, the risk function is a non-convex function that is minimized by the gradient descent method [11], which might cause trapping in local minima. Furthermore, in [11], the influence of several parameters was considered unimportant in building the model of a certain CF since it might cause noise to the decision making process of the classifiers. However, using only the important parameters may increase the possibility of wrenches and poses overlap from different CFs and in such cases the influence of the unimportant parameters would be necessary in computing the classification decision for each CF. In the scheme suggested throughout this paper, even though the partition matrix is computed using a derivative based optimization, finding the centers is performed through the GSA that would add more stochastic feature in minimizing the objective function given in (12). Furthermore, the LMS, used in computing the consequent parts parameters, was proved to be efficient in fitting the mapping of the TS fuzzy models on input- output data. This would significantly tune the whole mapping to fit more on the required input- output data. Moreover, using all available signals in the decision making process decrease the possibility of wrenches and poses overlap from different CF's. Now, if we examine the performance of both GS-FCA TS and SGB classifier of Fig. 3, the GS-FCA TS classifier is of superior performance due to the reasons mentioned above.

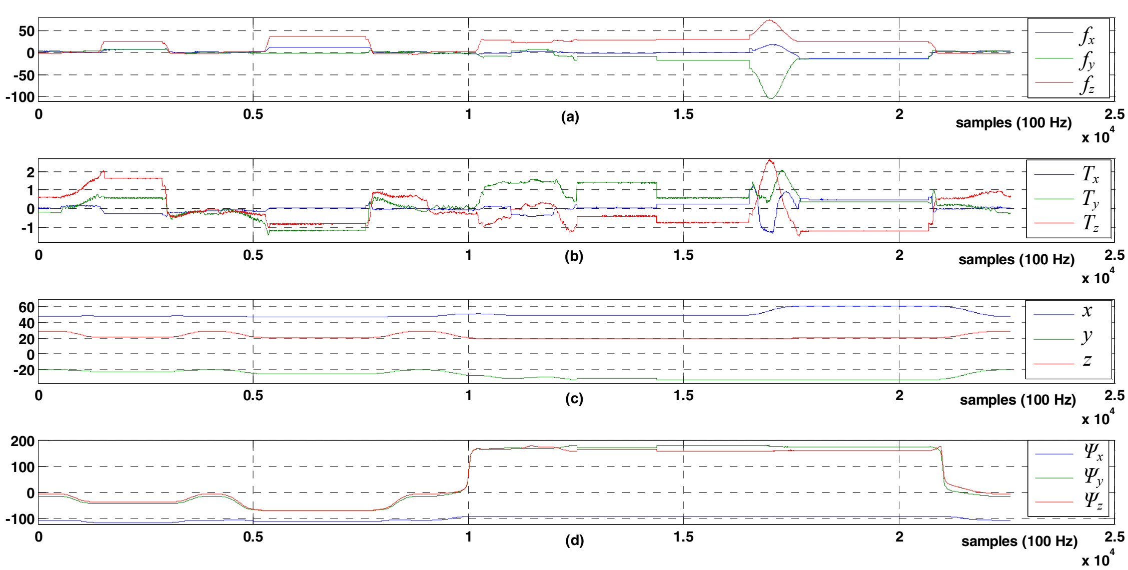


Fig. 2. The test phase signals (a) forces along the Cartesian axes (in N). (b) torques around the Cartesian axes (in N.m). (c) Cartesian position (in cm) (d) orientation around the Cartesian axes (in degree).

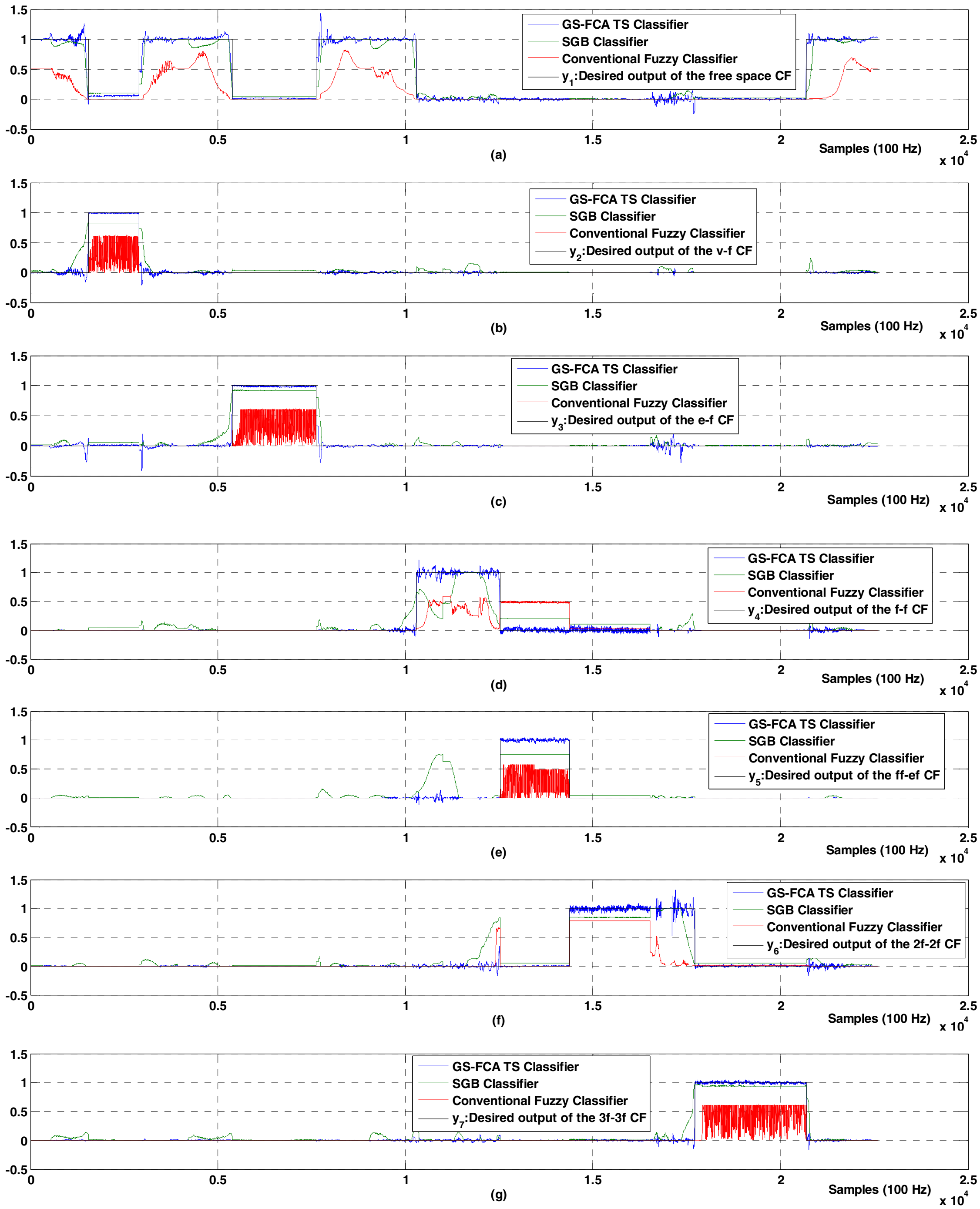


Fig. 3. The models outputs using the suggested GS- FCA TS Classifier , SGB Classifier , and the conventional fuzzy classifier (FC), and the desired output for (a) *free space* CF. (b) *v-f* CF. (c) *e-f* CF.(d) *f-f* CF. (e) *ff-ef* CF. (f) *2f-2f* CF. (g) *3f-3f* CF.

C4. In the approaches suggested in [13- 16], one if- then rule is dedicated for each CF that would reduce the mapping capability, especially for tasks with large number of CFs. Moreover, the mean and standard deviation of each signal are used in computing the center and width for each membership of the antecedent part of the *if-then* rules. The mean is very sensitive to data outlier possibly caused from human or sensors errors and consequently the fuzzy sets suggested in [13- 16] may not precisely quantify the captured signals. The consequent parts parameters are fixed to certain values that make the overall output to be bottle-necked by the imprecise antecedent parts membership functions. We suggested, in this paper, seven if- then rules for each CF to enhance the input-output mapping capability of each CF. The centers and width of each antecedent part membership function was computed using the GS-FCA which is more robust than the mean and

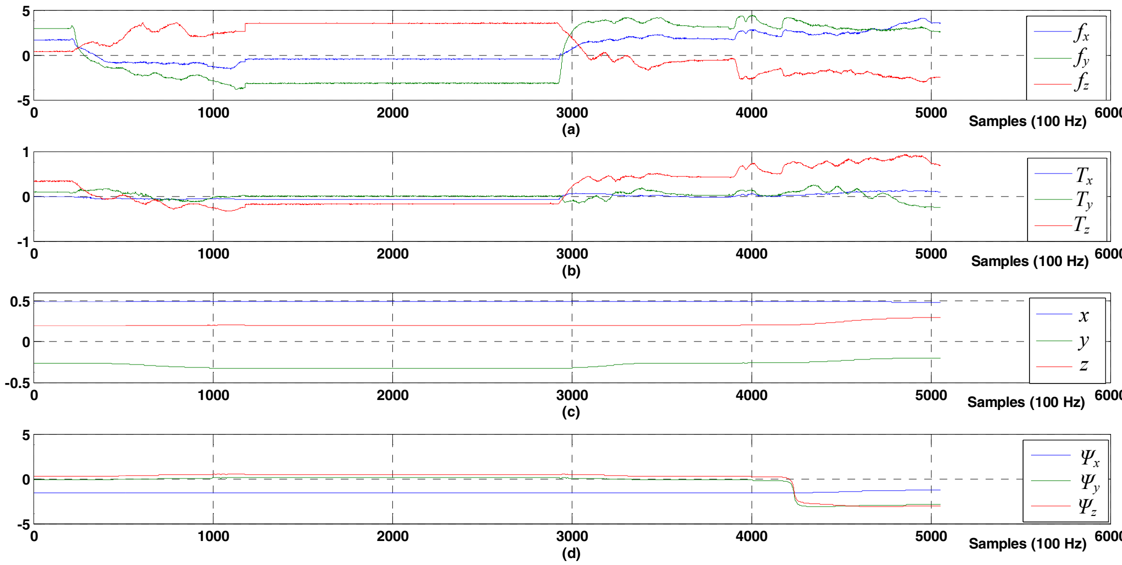


Fig. 4. The manipulated object captured signals for the second test task of sequence *(f-f)-(2f-2f)-(f-f)-(free space)* CFs (a) Cartesian forces in N (b) Torques around the Cartesian axes (in N.m) (c) Cartesian position (in m)(d) Orientation (in degrees).

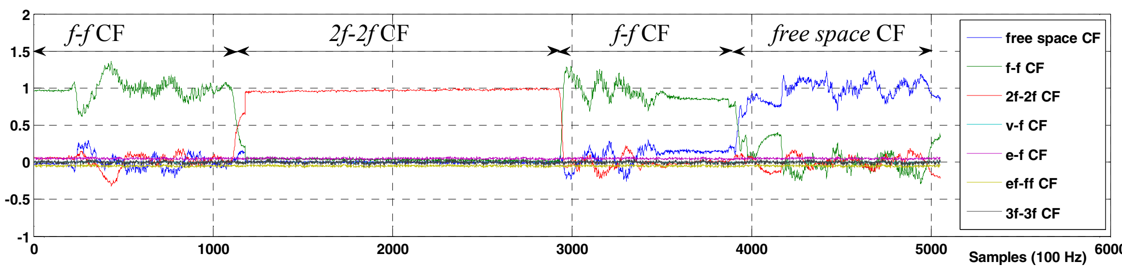


Fig. 5. The CF's models for the seocnd test task of of sequence *(f-f)-(2f-2f)-(f-f)-(free space)* CFs using the GS-FCA TS Classifier.

TABLE I. CLASSIFICATION SUCCESS RATE (%)

Classifier Type	***Classification Success Rate***
Conventional Fuzzy Classifier	38.5
SGB Classifier	81.2
GS-FCA TS Classifier	92.4

standard deviation against data outliers that are caused from human or sensors errors and enhanced signals representation by the fuzzy sets is obtained. Tuning the consequent part parameters added more flexibility to the mapping process of each model and small degradation in the antecedent part could be compensated by the LMS tuning of the consequent part parameters. Those three causes were the driving forces behind the superiority of the GS-FCA TS fuzzy classifier over the conventional fuzzy classifier (see Fig. 3).

C5. As a comparison between the suggested GS-FCA TS fuzzy classifier, and the available SGB and conventional fuzzy classifiers, we computed the classification success rate for each approach. The computation was based on the fact that the threshold of decision making for each CF model is 0.5, i.e. the if a CF output ≥ 0.5 then a decision is made on that CF. If more than two CFs have the outputs ≥ 0.5, then the decision will be made to the largest one. Table 1 shows the average classification rate for both tasks given in Fig. 3. The enhanced mapping tuning capability of the GS- FCA TS classifiers resulted in improved classification success rate.

V. CONCLUSION AND FUTURE WORKS

A contact state recognition system has been suggested for compliant motion robots. The suggested recognition system depends on the Gravitational Search- Fuzzy Clustering Algorithm (GS-FCA) in computing the antecedent part parameters and the Least Mean Square (LMS) in tuning the consequent part parameters of the *if*-then rules of each CF. The suggested approach was used to recognize different contacts of a force- controlled robotic system manipulating a cube rigid object and interacting with an environment composed of three orthogonal planes. Experimental results were carried out to validate the suggested approach. The experimental test stand was composed of a KUKA Light Weight Robot (LWR) manipulating a rigid cube object that interacts with an environment of three orthogonal planes. The robot was programmed by a human operator such that the cube makes different contacts with the environment. The wrench and pose readings were captured through the Fast Research Interface (FRI) available at the KUKA LWR with a sampling frequency of 100 Hz. Using the modeling scheme suggested throughout this paper, excellent performance was obtained in recognizing different CFs. The suggested approach was also compared with available approaches like the Stochastic Gradient Boosting (SGB) and conventional fuzzy classifiers. It was found that the suggested approach gives better performance in recognizing different CFs. Moreover, the requirement of knowing the CFs sequence of a task was successfully relaxed. Despite the given performance of the suggested scheme, the classification relied mainly on the LMS and fuzzy clustering in fitting the suggested models to the input-output data. A significant improvement can be obtained if the concept of the Fuzzy Maximum Likelihood (FML) is combined in in estimating the TS fuzzy classifier parameters. So, future works should focus on employing the FML based classification in recognizing CFs for compliant motion robots. Furthermore, evaluating the number of *if-then* rules for each CF model can be combined with the mapping efficiency and optimal number of *if-then* rules can be deduced according to a certain measure, like the deviance of Bernoulli or another measure that can assist in evaluating the mapping ability of a T-S fuzzy model to the input-output data.

ACKNOWLEDGMENT

The authors would like to thank Mr. Raphael Hinger and all the technical support staff at the University of Luxembourg-Campus Kirchberg for providing their support in building the test stand of the experiment presented in this paper.

REFERENCES

[1] R. S. Desai and R. A. Volz, "Identification and verification of termination conditions in fine motion in presence of sensor errors and geometric uncertainties," in *Proc. 1989 IEEE Int. Conf. Robotics and Automation*, Scottsdale, AZ, May 1989, pp. 800- 807.

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

[3] D. J. Austin, and B. J. McCarragher, "Model- adaptive hybrid dynamic control for robotic assembly tasks," *Int. J. Robot. Res.*, vol. 18, pp. 998- 1012, October, 1999.

[4] Y. Fei, and X. Zhao, "An assembly process modeling and analysis for robotic multiple peg- in- hole," *Jour. Intell. Robot. Syst.*, vol. 36 pp. 175- 189, 2003.

[5] G. E. Hovland, and B. J. McCarragher, "Hidden Markov models as a process monitor in robotic assembly," *Int. J. Robot. Res.*, vol. 17, no. 2, pp. 153- 168, February, 1998.

[6] H. Y. Lau, "A hidden Markov model- based assembly contact recognition," *Mechatronics*, vol. 13, pp. 1001- 1023, 2003.

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

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

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

[10] H. Okuda, H. Takeuchi, S. Inagaki, and T. Suzuki, "Modeling and analysis of peg-in-hole task based on mode segmentation," *2008 SICE Annual Conf.*, Tokyo, Japan, 20- 22 August, 2008, pp. 1595- 1600.

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

[12] K. Hertkorn, M. A. Rao, C. Preusche, C. Borst, and G. Hirzinger, "Identification of contact formations: resolving ambiguous force torque information," in *Proc. of the 2012 IEEE Int. Conf. Robot. and Autom.*, Saint Paul, Minnesota, USA, 14- 18 May, 2012, pp. 3278- 3284.

[13] M. Skubic and R. A. Volz, "Learning force sensory patterns and skills from human demonstration," in *Proc. of the 1997 IEEE Int. Conf. Robot. and Autom.*, New Mexico, 20- 25 April, 1997, pp. 284- 290.

[14] L. J. Everett, R. Ravuri, R. A. Volz, and M. Skubic, "Generalized recognition of single – ended contact formations," *IEEE Trans. Robot. and Autom.*, vol. 15, no. 5, pp. 829- 836, October, 1999.

[15] M. Skubic, S. P. Castrianni, and R. A. Volz, "Identifying contact formations from force signals: a comparison of fuzzy and neural network classifiers," in *Proc. of the 1997 Int. Conf. on Neural Networks*, Houston, TX, 9- 12 June, 1997, pp. 1623- 1628.

[16] M. Skubic and R. A. Volz, "Identifying Single- Ended Contact Formations from Force Sensor Patterns," *IEEE Trans. Robot. and Autom.*, vol. 16, no. 5, pp. 597- 603, October, 2000.

[17] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 26, issue 2, pp. 123- 140, August, 1996.

[18] A. Hatamlou, S. Abdullah, and H. Nesamabadi- pour, "Application of gravitational search algorithm on data clustering," *Rough Sets and Knowledge Technology- Lect. Notes on Comput. Sci.*, vol. 6954, pp.337- 346, 2011.

[19] C. Li, J. Zhou, B. Fu, P. Kou, and J. Xiao, "T-S fuzzy model identification with a gravitational search based hyperplane clustering algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 2, pp. 305- 317, April, 2012.

[20] J. T. Spooner, M. Maggiore, R. Ordonez, and K. M. Passino. *Stable Adaptive Control and Estimation for Nonlinear Systems: Neural and Fuzzy Approximator Techniques*. John Wiley & Sons, 2002.

[21] S. Miyamoto, H. Ichihashi, and K. Honda. *Algorithms for Fuzzy Clustering:Methods in c- Means Clustering with Applications*. Springer, 2008.

[22] T. A. Runkler, and C. Katz, "Fuzzy clustering by particle swarm optimization," in *Proc. of the 2006 IEEE Int. Conf. on Fuzzy Syst.*, Vancouver, BC, Canada, 16- 21 July, 2006, pp. 601- 608.

[23] E. Rashedi, H. Nesamabadi- pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232- 2248, 2009.

[24] E. Kim, M. Park, S. Ji, and M. Park, "A new approach to fuzzy modeling," *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 3, pp. 328–337, August, 1997.

[25] J. Abonyi, R. Babuška, and F. Szeifert, "Modified Gath- Geva fuzzy clustering for identification of Takagi- Sugeno Fuzzy Models," *IEEE Trans. Syst. Man. Cyb. Part B:Cyb.*, vol. 32, no. 5, pp. 612- 621, Oct., 2002.