

A modular extreme learning machine with linguistic interpreter and accelerated chaotic distributor for evaluating the safety of robot maneuvers in laparoscopic surgery

Ahmad Mozaffari, Saeed Behzadipour*

Sharif University of Technology, Azadi Street, Tehran, Iran

ARTICLE INFO

Article history:

Received 8 March 2014

Received in revised form

1 September 2014

Accepted 2 October 2014

Communicated by G.-B. Huang

Keywords:

Laparoscopic surgery

Medical robotics

Soft tissue modeling

Clustering

System identification

Fuzzy inference system

ABSTRACT

In this investigation, a systematic sequential intelligent system is proposed to provide the surgeon with an estimation of the state of the tool-tissue interaction force in laparoscopic surgery. To train the proposed intelligent system, a 3D model of an *in vivo* porcine liver was built for different probing tasks. To capture the required knowledge, three different geometric features, i.e. Y displacement of the nodes on the upper surface and slopes on the closest node to the deforming area of the upper edge in both X-Y and Z-Y planes, were extracted experimentally. The numerical simulations are conducted in three independent successive stages. At the first step, a well-known partition-based clustering technique called accelerated chaotic particle swarm optimization (ACPSO) is used to cluster the information of database into a number of partitions. Thereafter, a modular extreme learning machine (M-ELM) is used to model the characteristics of each cluster. Finally, the output of M-ELM is fed to a Mamdani fuzzy inference system (MFIS) to interpret the safety of robot maneuvers in laparoscopic surgery. The proposed intelligent framework is used for real-time applications so that the surgeon can adjust the movements of the robot to avoid operational hazards. Based on a rigor comparative study, it is indicated that not only the proposed intelligent technique can effectively handle the considered problem but also is a reliable alternative to physical sensors and measurement tools.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

In robotic laparoscopic surgery, robots are used to move the laparoscopic instruments that are inserted into the body through small incisions. The movements of robots are controlled with the aid of a camera [1]. The robotic system comprises a master robot, a slave robot and a communication channel. Generally, the master robot has two different obligations: (1) receiving the movements from surgeon's hand and (2) simulating the tool-tissue interaction forces on the surgeon's hand. The latter characteristic of master robot enables the surgeon to feel the tool-tissue reactions over the surgery. The second robot, known as the slave robot transfers the surgeon's motions to the surgery instruments working inside the patient's body. As a master-slave system, these two independent robots are connected by a communication channel. Indeed, this communication hardware provides a link between the surgeon and the surgery tools. During the operation, surgeons observe the slave robot movements through a 2D or 3D video system, and

discern the next motions of the robot. Fig. 1 depicts the block diagram of a robotic laparoscopic system.

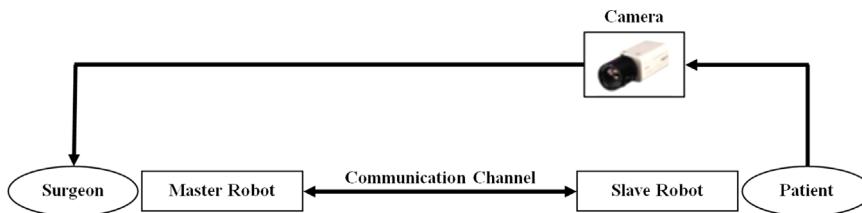
Robotic laparoscopic surgery has several prominent advantages. The most salient assets of laparoscopic surgery are:

- (1) The use of robot which in turn causes less damages and hazards by filtering the sudden movements as well as vibrations of the surgeon's hand.
- (2) Fast recovery and low risk of complications such as infection.
- (3) Using small incisions for surgery which in turn decreases the damages to the patient's body.
- (4) Reducing the need for repeated surgery [2].

In spite of the abovementioned benefits, in most of the robotic laparoscopic surgery systems, the surgeon does not have any tactile perception, and thus cannot evaluate the tool-tissue force [3]. This is while it is crucial to surgeons to have a sense from tool-tissue interactions to verify the exerted forces on patient's tissues by the grasper. One way to address this problem is using sensorized tools to measure the forces directly which will be then recreated at the master robot using a force control scheme. However, due to the size

* Corresponding author. Tel.: +98 2166165542.

E-mail address: behzadipour@sharif.edu (S. Behzadipour).

**Fig. 1.** The block-diagram of a robotic laparoscopic system.

and sterilizability requirements, design of such intricate sensors with high SNR is challenging [4].

The second way is to contrive proper algorithmic numerical methods to estimate/evaluate the tool-tissue forces based on visual data. In such frameworks, for instance, some geometric features are extracted from the real-time images taken from the surgery site. Those pieces of information are then used to develop numerical physic-based or soft models. Once the model is prepared, it can be used for estimating the tool-tissue interaction forces.

The physic-based nonlinear modeling of soft tissues and numerical methods such as boundary element and finite element (FE) are widely used to provide precise pragmatic estimators. Tirehdast et al. [2] used a nonlinear FE method to investigate the tissue deformation and sliding occurrence between a three-fingered grasper and human spleen. Basafa and Farahmand [5] simulated the nonlinear visco-elastic deformations of soft tissue for real-time applications. Miller [6] used FE approach to develop a model of brain tissue to be used in surgical procedure. The numerical results were compared with analytical solutions to ensure the validity of solutions obtained by FE. Nienhuys and Van Der Stappen [7] used an adaptive version of FE technique and quasi-static stick slip friction concept for simulating needle-tissue interactions over the surgery. By further exploration of archived literature, it can be easily seen that numerical modeling methods such as FE were widely used in medical robotic to model different types of tool-tissue interactions. However, although such models yield relatively precise results, their practical applications are still limited due to the expensive and time consuming computational requirements. In fact, such techniques cannot be used for real-time applications unless several simplifications are exerted on their structure which in turn reduce their precision [4,8].

The need for fast and inexpensive computational techniques persuaded the researchers of medical robotic society to take the advantages of computational intelligence (CI) and surrogate expert systems. Generally, intelligent approaches have the following advantages as compared to numerical approaches such as FE:

- (1) Their training does not require any knowledge about the physics of the problem.
- (2) They are really inexpensive computational approaches. There is neither a need for discretizing the solution domain nor solving a system of equations. Rather, they combine a set of soft computational units (rules in fuzzy systems and neurons in neural network) to model any nonlinear system.
- (3) They are suited for real-time applications and incremental learning. This feature enables the experts to use the experimental feedbacks to retrain or refine the structure of such modeling approaches over time. This is while using adaptive FE methods significantly increases the algorithmic complexity and computational time.

Despite such prominent advantages, CI modeling techniques have not found their real reputation in medical robotic applications such as laparoscopic surgery. There are relatively few seminal researches in literature trying to confirm the applicability of CI techniques in laparoscopic surgery. Gholipour et al. [9] used artificial neural

network (ANN) to identify the conversion of laparoscopic cholecystectomy to open surgery. Their experiments demonstrated that neural modeling can yield reliable predictive results based on the preoperative health characteristics of patients. Greminger and Nelson [10] used a neural network together with images of an elastic rubber torus deformed under compressive load to estimate the applied force over the laparoscopic surgery. After experiments, they observed that the soft modeling tool can effectively identify the tool-tissue force. Becker et al. [11] fused the fuzzy and neural computational principles to develop a mapping scheme suitable for evaluating pneumoperitoneum in laparoscopic surgery. Their findings endorse the better performance of neuro-fuzzy modeling tool as compared to statistical methods. Estebanez et al. [12] used ANN and hidden Markov models to recognize the movements of two-arm-surgical robotic system for laparoscopic surgery. The simulation results revealed that the developed intelligent model is capable of correctly recognizing and distinguishing certain standard surgical maneuvers. Kohani et al. [4] utilized a feed forward neural network and geometrical features obtain from 2D images to estimate the tool-tissue force in laparoscopic surgery. Mozaffari et al. [13] proposed two hybrid neuro-evolutionary fuzzy systems and a synchronous self-learning hyper level supervisor to identify the tool-tissue force in laparoscopic surgery.

In spite of novelty and practical innovations in all abovementioned proposals, CI techniques have not been used for automate decision making over the procedure. To be more to the point, identification of interactions between surgery instruments and patient issues may not always be interpretable for surgeon. To tackle the problem, it is really essential to use a consequent linguistic inference soft method which helps the surgeon to conveniently evaluate the quality of operation. However, to the best knowledge of the authors, such requirement was not clearly fulfilled. In a preliminary research by Huang et al. [14], the potential of fuzzy classifier for evaluating the possibility of classifying skill levels was examined. To do so, some scoring metrics such as the number of errors made and the economy of movement were taken into account. Different operators, i.e. laparoscopic surgeons, surgical assistance and non-surgical staffs were assigned to perform certain operations by robots. The results were not so promising. Based on experiments, the authors reported that their effort for developing an expert system capable of providing a qualitative feedback from the operation did not yield reliable and conclusive results. They stated a number of reasons, and outlined some areas for future works to augment the reliability of their expert system. In a seminal research, Feng et al. [15] developed a fuzzy based decision making expert system to help the surgeon to assess the success of operation. However, on that research, micro sensors were used to provide information for fuzzy inference system (FIS). Therefore, the methodology proposed by Feng et al. [16] may not be a deliberate strategy for all types of minimum invasive surgery. In this investigation, instead of using physical micro sensors, an efficient neural modeling technique (as a data-driven soft sensor) is taken into account.

The proposed expert system in this paper has two main obligations, i.e. tool-tissue force estimation and linguistic interpretation. Although the first objective, i.e. soft modeling, was addressed before, there is no effort for increasing the training speed of the existing soft techniques. This is while contriving fast and reliable

soft identifier is of high importance. This is because using a fast identifier enables us to easily extend the realm of machine learning to incremental learning concepts. To address such an important issue, the authors take advantage of modular extreme learning machine (M-ELM). M-ELM is a modular version of extreme learning machine (ELM) network in which a number of sole ELMs are combined to improve the accuracy and generalization of prediction [17,18]. The main provocation behind using a linguistic-based inference system (second phase of the proposed expert system) is to aid surgeon to conveniently evaluate/ interpret the safety of the operation and avoid hazardous motions over the surgery. Generally, main motivations behind the integration of the FIS decision making mechanism and soft modeling technique are twofold:

- (1) As the FIS is a soft computation framework, it can be neatly hybridize with neural identifier. In other words, the outputs of M-ELM can be easily used for uncertain inference in fuzzy part.
- (2) Integration of those two soft techniques enables us to develop a fully automatic expert system which in turn decreases the risk of complications such as infection and severe tissue damages.

Besides, the main motivations of proposing the current sequential intelligent framework are as follows:

- (1) For the first time, the computational capabilities of ELM is probed for designing a fully automate linguistic inference tool for assessing the safety of robot maneuvers in laparoscopic surgery. The main motivation behind the use of ELM emanates from the urgent need for devising a computationally efficient intelligent interpolator at the heart of the sequential linguistic inference tool which not only can be used as a precise identifier, but also has a significant capability to be-retrained over the surgery, if required. The necessity of having a fast and accurate identifier is obvious, especially for applications in which any lag on the performance of the tool-tissue force estimator may result in life hazards. It is indicated that by using M-ELM, a very fast and accurate intelligent tool can be developed which can be reliably used for online applications.
- (2) Through the simulations, the authors demonstrate the high potential of ELM to simultaneously be fused with clustering and inference systems. Indeed, the current research aims at taking a stride toward designing data-driven linguistic inference tools which can be used for online analyzing and processing of large-scale nonlinear laparoscopic databases.

The rest of the paper is organized as follows. Section 2 is devoted to experimental procedure required for gathering the database. The description of the proposed expert identification-interpretation

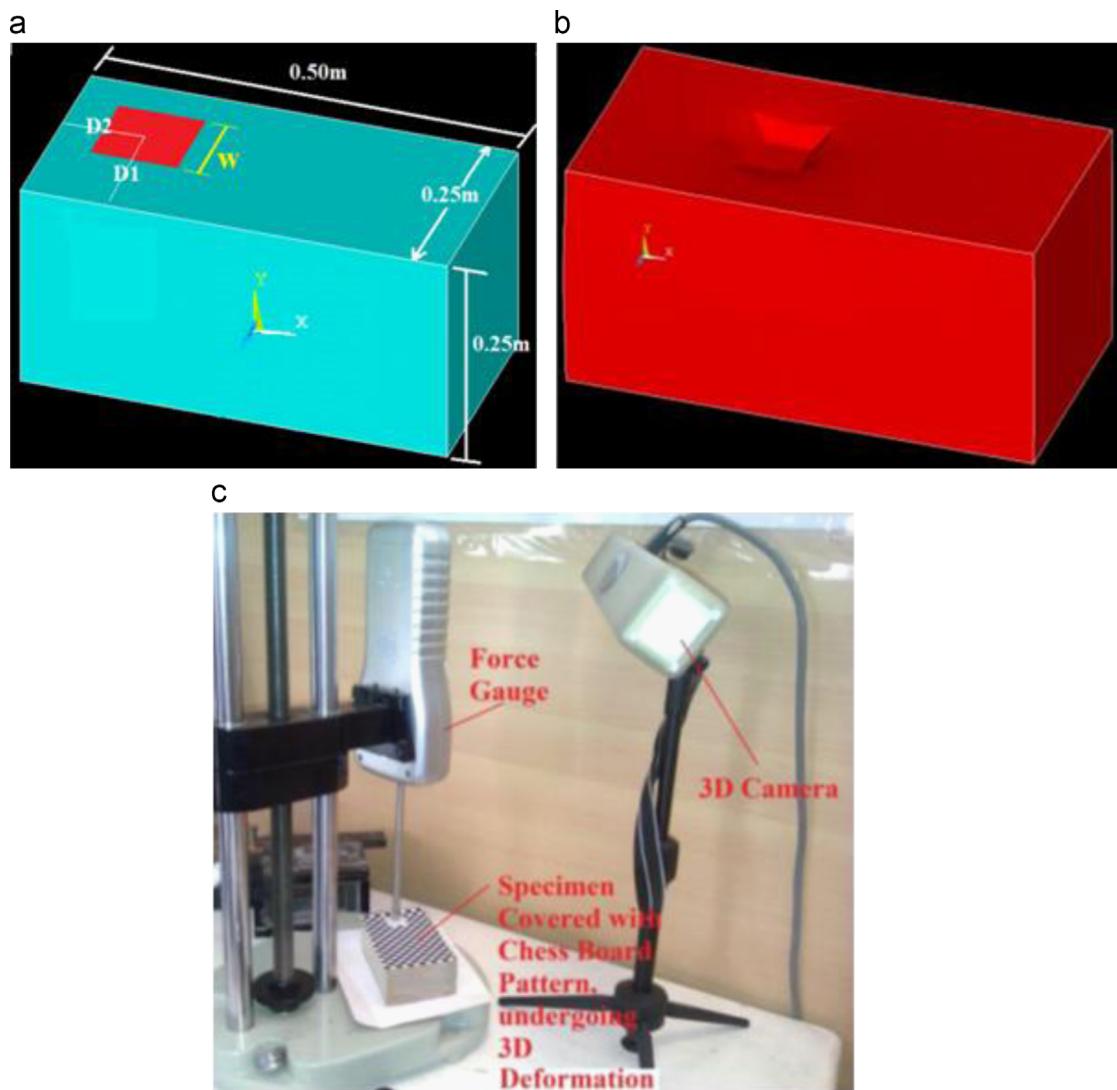


Fig. 2. (a) characteristics of the displacement constraint, (b) deformed sample, (c) the experimental setup.

system is given in [Section 3](#). The results are presented in [Section 4](#). Finally, the paper is concluded in [Section 5](#), and some potential investigation topics for future works are presented.

2. Problem description

To conduct the experiments, a $0.5 \times 0.25 \times 0.25$ cube with hyper-elastic mechanical properties of pig's liver was built, and simulated in ANSYS software [19]. The values of hyper-elastic coefficients of the considered *in vivo* pig liver were extracted from Lister et al. [20]. [Fig. 2\(a\)](#) indicates the characteristics of axis and exertion of displacement constraint on the upper side of the cube. The small red square on the upper side of the cube shows the region in which the Y-displacement constraint is exerted. Obviously, determining the position of the center of the square (D_1 and D_2), the length of the square (W) and the amount of displacement constraint are of high importance for analysis. Following additional constraints should be defined to avoid impractical placement of the displacement square:

$$D_2 + \left(\frac{W}{2}\right) + 1.5W < 0.5 \quad (1)$$

$$D_1 + \left(\frac{W}{2}\right) + 1.5W < 0.25 \quad (2)$$

After determining the place of displacement square, the cube is meshed with element types of SOLID 187. For higher precision and better analysis, we use smaller meshes in the upper side of the cube and in vicinity of the displacement square. By conducting the mesh-based simulation, we obtain the gripper force and the maximum local stress at each node for certain characteristics of the displacement square. [Fig. 2\(b\)](#) indicates the deformation of the cube for certain characteristics of the displacement square. [Fig. 2\(c\)](#) indicates the test facility used to collect the experimental data. The experiments were conducted for 2115 different deformation characteristics of the cube. To extract the most important information for training the expert system, the authors conducted a geometrical feature selection experiment. To do so, initially, a set of potential features were verified and then the correlation function was used to discard the redundant ones. After the termination of the process, it was observed that Y-displacement of the nodes on the upper surface (inp_1) and slopes on the closest node to the deforming area of the upper edge in both X-Y and Z-Y planes (inp_2 and inp_3 , respectively) are considered as final geometrical features. [Fig. 3](#) indicates the variations of the geometrical features for all 2115 experimental data. It was observed that the final

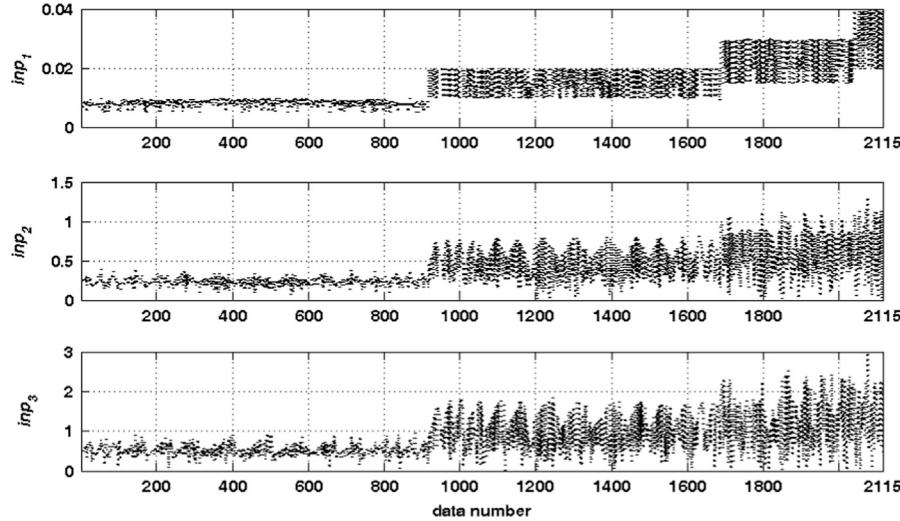


Fig. 3. Variation of the geometrical features for the experimental data.

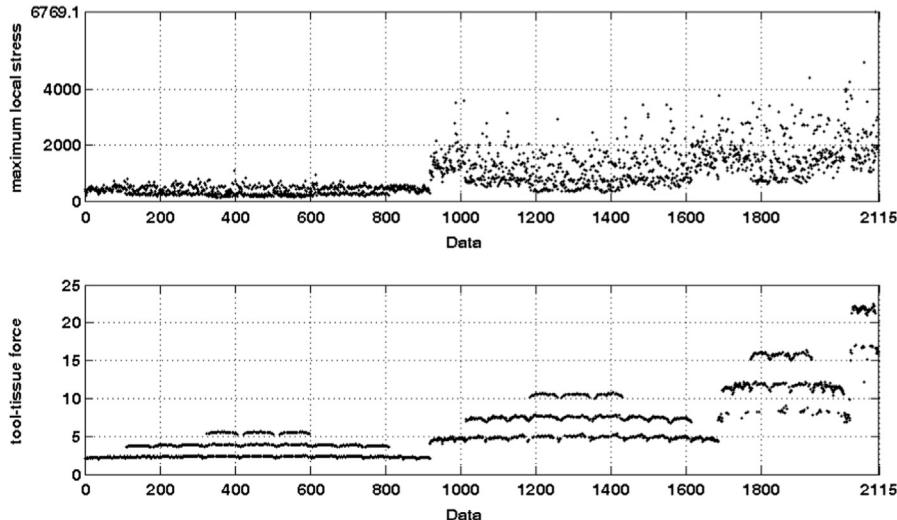


Fig. 4. The values of the tool-tissue force and the maximum local stress.

geometrical features have indeed physical interpretations which naturally correlate with the mechanics of the tissue. Additionally, the use of such features relaxes the authors from tracking all points on the edges of the body. The corresponding values of the tool-tissue force and the maximum local stress are given in Fig. 4. Those pairs of input-output data are used to train the proposed expert system.

3. Methods

As it was mentioned, the proposed intelligent expert system combines two different soft approaches for both identification and interpretation tasks. In the rest of the section, these two systems are scrutinized.

3.1. Modular extreme learning machine neural network

The modular extreme learning machine is developed based on the concepts of ELM and modular architecture design. The system also requires an intelligent algorithm verifying that which data should be forwarded to which module. The details of the developed network are given in the rest of this subsection.

3.1.1. Extreme learning machine

ELM is a neural learning algorithm without iteration in which there is no need for adjusting the input weights of ANN [21]. The main motivation behind the proposition of such a learning technique is to increase the learning speed of ANN. ELM strategy enables us to tune the ANN hundred times faster than what is required for gradient descend learning algorithms. Besides, it was shown that in most cases, it yields much more promising results as compared to other deep-seated learning strategies [21]. Since its proposition by Huang et al. [17], it has been successfully applied to different numerical problems, and its theory has been clearly investigated, extended and established in literature. Huang et al. [21] comprehensively surveyed the chronological development, advantages and disadvantages of ELM-based soft systems. Feng et al. [16] proposed an error minimized ELM with growth of hidden nodes. It was observed that the method can be used effectively for incremental learning tasks. Huang and Chen [22] investigated the impact of random search-based ELM for incremental learning. It was observed that such a strategy has a significant effect on decreasing the complexity of the ANN. Huang and Chen [23] also modified the capabilities of incremental ELM technique by using a convex optimizer for re-evaluating the output weights of hidden nodes. Wang et al. [24] proposed a modified localized generalization error model with a novel architecture selection algorithm in which the training was conducted by ELM technique. Kapil [25] extended the application of ELM to ϵ -insensitive error loss function-based regression problem. After comparative study, it was observed that the resulted method outperforms the support vector regression (SVR) system.

By exploring the archived literature, it can be easily inferred that ELM has found its reputation and applicability in recent decades, and several theoretical rationales have been provided to extend its characteristics. However, until recent years, there have not been many reports in the literature addressing the use of ELM-based ANN for important real-life applications. A research by Yeu et al. [26] is among the primary works in which ELM was used for terrain reconstruction problem. After that, ELM-based ANNs were used for microarray gene expression cancer diagnosis, human action recognition, speaker recognition, face recognition, etc. [21]. The success of ELM based ANNs in a wide spectrum of real-world applications demonstrates their authentic performances. In this work, we extend the application of ELM to the realm of laparoscopic surgery.

Generally, ELM is trained analytically based on a number of algebraic equations. Consider that N training samples $(\mathbf{x}_i, \mathbf{t}_i)$ are available where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{t} \in \mathbb{R}^m$ represent the input and target vectors, respectively. Let us consider that ANN has \tilde{N} neurons in its hidden layer. It was rigorously proved that for infinitely differentiable activation function, the number of the required hidden nodes \tilde{N} for training N data points is less than or equal to N . For a single hidden layer ANN, the neural map for transforming the input vector to the output one can be formulated as:

$$\sum_{i=1}^N W_i^{HO} g(W_i^{IH} \tilde{\mathbf{x}}_j + b_i) = o_j, \quad j = 1, \dots, N \quad (3)$$

where $W_i^{IH} = [w_{i,1}, w_{i,2}, \dots, w_{i,n}]^T$ represents the synaptic weight vector connecting the input nodes to the i th hidden node, $W_i^{HO} = [\beta_{i,1}, \beta_{i,2}, \dots, \beta_{i,\tilde{N}}]^T$ is the weight vector connecting the i th hidden node to the output nodes, and g is a continuous activation function (*logsig* in our case). Using pseudo-inverse technique enables us to analytically tune the parameters of W_i^{HO} so that the network estimates the outputs of each input vector with zero error, i.e. $\sum_{j=1}^N \|o_j - t_j\| = 0$. As it was mentioned, in ELM, there is neither a need for tuning the values of the input-hidden weights nor the biases of the hidden nodes. As a result, for any random values of W_i^{IH} and b_i , there exists optimum values for vector W_i^{HO} which yield the following equilibrium:

$$\sum_{i=1}^N W_i^{HO} g(W_i^{IH} \tilde{\mathbf{x}}_j + b_i) = t_j, \quad j = 1, \dots, N \quad (4)$$

To perform the algebraic calculations, we compact the above system of equations into *hidden layer output matrix* H as:

$$H W^{HO} = T \quad (5)$$

where

$$H = \begin{bmatrix} g(W_1^{IH} \times \mathbf{x}_1 + b_1) & \dots & g(W_{\tilde{N}}^{IH} \times \mathbf{x}_1 + b_{\tilde{N}}) \\ \vdots & \dots & \vdots \\ g(W_1^{IH} \times \mathbf{x}_N + b_{\tilde{N}}) & \dots & g(W_{\tilde{N}}^{IH} \times \mathbf{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}},$$

$$T = \begin{bmatrix} t_{1,1}^T & \dots & t_{1,m}^T \\ \vdots & \dots & \vdots \\ t_{N,1}^T & \dots & t_{N,m}^T \end{bmatrix}_{N \times m}, \quad W^{HO} = \begin{bmatrix} W_{1,1}^{HO} & \dots & W_{1,m}^{HO} \\ \vdots & \dots & \vdots \\ W_{\tilde{N},1}^{HO} & \dots & W_{\tilde{N},m}^{HO} \end{bmatrix}_{\tilde{N} \times m}$$

To tune the values of W^{HO} , we need to find the least square solution of the general system given in Eq. (5) which yields the smallest training error (\hat{W}^{HO}):

$$\|H \times \hat{W}^{HO} - T\| = \min_{W^{HO}} \|H \times W^{HO} - T\| = \|H \times H^\dagger \times T - T\| \quad (6)$$

where H^\dagger is the Moore-Penrose generalized inverse of matrix H [27]. As it can be seen, ELM training strategy is quite straightforward. However, finding the inverse of matrix H may cause some problems in practice. Several techniques such as orthogonal projection, orthogonalization, singular value decomposition (SVD) and iterative techniques were proposed to calculate the Moore-Penrose inverse. SVD technique has been proven to be the better one as it can be used for calculating the inverse of a wide spectrum of matrices. To be more to the point, in a seminal paper by Huang et al. [17], it was demonstrated that unlike orthogonal projection technique, SVD can significantly increase the stability of the ELM, as it does not suffer from the numerical difficulties associated with asymmetric and ill-conditioned matrices. Besides, the computational complexity of SVD for calculating the inverse of a matrix with $N \times \tilde{N}$ arrays is about $4N\tilde{N}^2 + 8\tilde{N}^3$ which is trivial for a relatively small matrix like H [39]. In the light of such consideration, the authors calculate the inverse of matrix H using SVD technique. Let

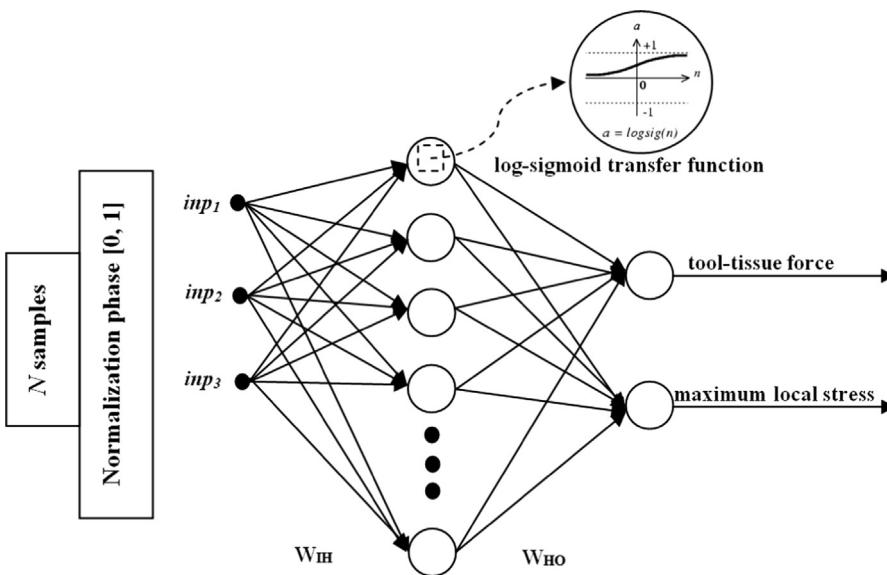


Fig. 5. Structure of the ELM-based neural network with a *logsig* activation function.

us create a factorization form for matrix H as:

$$H = U\Sigma V^* \quad (7)$$

where U is a $N \times N$ unitary matrix, Σ is a $N \times \tilde{N}$ rectangular diagonal matrix with nonnegative real numbers on the diagonal, and V^* is a $\tilde{N} \times \tilde{N}$ unitary matrix. Now, the inverse of matrix H can be calculated as:

$$H^\dagger = V\Sigma^\dagger U^* \quad (8)$$

where V and U^* are the conjugate transpose of unitary matrices V^* and U , respectively, and Σ^\dagger is derived by replacing every non-zero diagonal array by its consequent reciprocal and transposing the resulted matrix.

Fig. 5 indicates the structure of ELM-based ANN. Generally, ELM has several prominent advantages as compared to traditional learning algorithms. The most important features of ELM are (1) low computational cost, (2) analytical tuning of ANN's structure, (3) ability to be combined with hyper-level supervisors/optimizers, (4) robustness, (5) acceptable prediction accuracy, and (6) generalization.

Such promising characteristics have persuaded the authors to adopt ELM for fast and effective learning of ANN. This is because having a fast adjustable ANN provides us with good opportunity to mitigate the computational efforts and also use it for incremental identification in laparoscopic operation. However, in spite of the aforementioned characteristics, ELM-based ANN has its own practical limitations. The main drawback is the weak performance of ELM for large amount of data. This is because for tuning the hidden-output weights of ANN, we need to perform algebraic operations, such as calculating matrix inverse. Therefore, although we are using SVD technique, there is a possibility for failure due to singularity issues. As it was mentioned, the number of data in our database is relatively large. Therefore, it would be rational to take advantage of modular computation to alleviate the difficulties associated with extracting knowledge from large databases [28].

Modular neural network is an ANN which includes a series of independent networks [29]. Each of those sole networks (known as module) is assigned to fulfill a sub-task by operating on separate portion of inputs. By combining the outcome of independent modules, the modular network is able to complete the entire task. The idea of modular neural networks was inspired by segmentation and modularization found in the brain. The main motivation behind the use of modular networks is to reduce a large and complex neural

network to smaller and much more manageable components. As the modular approach breaks the task to a set of sub-tasks, we can easily manage it to complete its objective with high accuracy and robustness. One of the interesting characteristics of any modular framework is fostering the idea of parallel computing. To be more to the point, even if the considered modules fulfill their objective independently, the output of the whole system follows a unified stream line. Therefore, modular frames suggest the integration of simple, fast, and parallel independent sub-processors for solving tedious and complex problems [30].

Since its proposition, modular soft computing concept has been used for a wide range of practical applications including medical systems [31]. The results proved that modular computing not only inherit the advantages of their constructive modules but also augment their efficiency, accuracy and robustness while decreasing the complexity of information processing. Taken the promising results of modular soft computing into account, the authors propose a modular extreme learning machine (M-ELM) to extract knowledge from the provided database. The modular identifier is constructed from a number of independent ELMs. Each of these ELMs (modules) aims at processing a predefined part of the database. Obviously, to provide the condition for modular learning, we need to divide the database to a number of sub-databases and assign a module to cope with each of them. The process is known as '*distribution of information*' and the task is done by a '*distributor*'. Clustering techniques can be really effective for distribution process. It is only needed to define a set of metrics and assign the partitioning process to clustering approaches. After producing a set of predefined clusters, they are fed to independent modules (ELMs) so that each module can extract the required knowledge. Finally, by integrating the results of independent modules, the modular frame can effectively extract the entire knowledge from complete database, and consequently identify the tool-tissue force and maximum local stress.

3.1.2. Accelerated chaotic particle swarm optimization

Swarm and evolutionary metaheuristic algorithms can be effectively used for clustering tasks [32]. There are so many swarm-based methods in literature proposed for different type of clustering problems. When using them, two crucial metrics should be evaluated: (1) computational efficiency and (2) high performance. Proposed by Chuang et al. [33], accelerated chaotic particle swarm optimization (ACPSO) is a recent spotlighted clustering algorithm

that operates based on the combination of chaotic search and swarm intelligent based exploration/exploitation. It was experimentally proved that ACPSO can outperform a wide range of partitioning based clustering techniques such as *K*-Means, PSO, NM-PSO, K-PSO, K-NM-PSO, CPSO and GA [33,34]. Given the results of those experiments, we adopt ACPSO and use it as distributor in our expert model. Besides, to attest the authenticity of conclusions presented by Chuang et al. [33], the performance of ACPSO is compared with a number of rival clustering techniques for partitioning our database. Similar to the original paper, instead of random walk, *logistic map* is used to sequentially update the position of particles in a diversified fashion. Detailed discussions on the characteristics of ACPSO were given in the original paper [33]. For the sake of brevity, in this paper, we just provide a stepwise definition for ACPSO.

Step 1: Initialization: Stochastically initialize the position ($x = [x_1, x_2, \dots, x_N]$) and velocity ($v = [v_1, v_2, \dots, v_N]$) of N particles within the solution space. Each particle contains the center position of all clusters. For clustering a data base of d ($i=1,\dots,d$) dimensions into k partitions ($j=1,\dots,k$), each particle should have $k \times d$ variables. It is worth mentioning that for such a problem, $3 \times k \times d$ particles are used to complete the task within $10 \times k \times d$ iterations.

Step 2: Acceleration: Before proceeding with PSO's position updating procedure, select one third of the particles for a *K*-Means like operation which is known as acceleration strategy. In this phase, firstly, each of the selected particles tries to cluster all of the data in database. This is done by assigning all data to the closest cluster (by calculating the Euclidian distance from the respective cluster center). This can be expressed as:

$$Euc_dist(x_p, c_j) = \sqrt{\sum_{i=1}^d (x_{p,i} - z_{j,i})^2} \quad (9)$$

where c_j denotes cluster center j , $z_{j,i}$ indicates the i th dimension of the center position of cluster j , $x_{p,i}$ shows the i th dimension of the p th data in the database, and $Euc_dist(x_p, c_j)$ represents the Euclidian distance between the j th centroid and the p th data. Next, for each particle in the acceleration phase, the center position of each cluster j is updated by calculating the mean position of all data belonging to cluster j . The procedure is mathematically expressed as:

$$z_j = \frac{1}{n_j} \sum_{\forall x_p \in c_j} x_p \quad (10)$$

where n_j indicates the number of data in cluster j , and $\forall x_p \in c_j$ represents the data points in cluster j .

Step 3: Partition based clustering: Assign all particles in the PSO to cluster all data in the database using Eq. (9).

Step 4: Fitness evaluation: Calculate the fitness of all particles using the sum of intra-cluster distances as:

$$fitness = \sum_{j=1}^k \sum_{\forall x_p \in c_j} \|x_p - z_j\| \quad (11)$$

where z_j represents the center position of the j th cluster and x_p represents data in that cluster. Obviously, the lower the fitness value, the better the clustering scheme.

Step 5: Update *gbest* and *pbest*: After evaluating the fitness of all solutions, the *gbest* and *pbest* are updated.

Step 6: Position and velocity update: Update the velocities and positions of all particles as (at iteration t):

$$Cr(t) = K \times Cr(t-1) \times (1 - Cr(t-1)) \quad (12)$$

$$v_{i,h}(t) = \omega \times v_{i,h}(t-1) + c_1 \times Cr(t) \times (pbest_{i,h} - x_{i,h}(t-1)) + c_2 \times (1 - Cr(t)) \times (gbest_h - x_{i,h}(t-1)) \quad (13)$$

$$x_{i,h}(t) = x_{i,h}(t-1) + v_{i,h}(t) \quad (14)$$

where h represents the h th dimension of each particle, K is equal to 4 and both c_1 and c_2 are equal to 2. It should be noted that the initial value of $Cr(0)$ can be any value spanning the unity except {0, 0.25, 0.5 and 0.75}.

Step 7: Stopping criterion: Terminate the procedure if the stopping criterion is met; otherwise, repeat steps 3 to 6.

The above mentioned procedure enables us to successfully implement the first phase of the proposed expert system. The resulted M-ELM can correctly identify the values of tool-tissue force and maximum local stress.

3.2. Fuzzy interpretation

At the second phase, the expert knowledge is used to provide a fuzzy inference system of Mamdani type (MFIS) which interprets the quality of robot maneuvers based on the outputs of the M-ELM. The main provocation of using MFIS is twofold: in one hand, fuzzy programming provides us with good opportunity to mitigate the undesired effects of noises and uncertainties, on the other hand, as a Mamdani system, the devised tool is able to yield its outputs in a linguistic form which is easier to comprehend for the surgeon. Mamdani fuzzy inference system has been successfully used for a wide range of engineering applications such as system identification and control [35]. MFIS is able to act as a linguistic-based decision maker capable of reconciling the dynamic environmental changes. We only need to train it based on the knowledge of experts and then

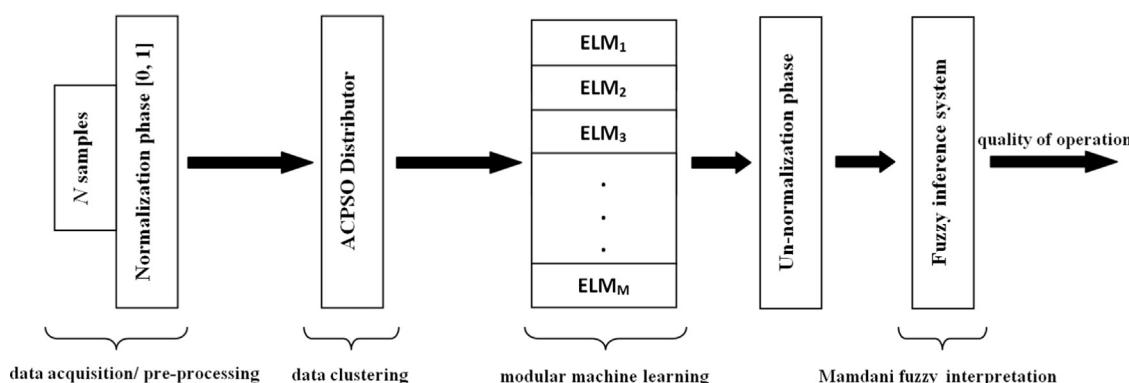


Fig. 6. Schematic illustration of the proposed expert system.

use it as a real-time interpreter, controller or identifier. Such a promising capability instigates the authors to construct a MFIS interpreter based on the knowledge of expert (laparoscopic surgeon and expert engineers), and mount it on the proposed intelligent system to capture the quality of robot maneuvers in laparoscopic surgery. To implement the MFIS, following steps should be taken:

Step 1: Fuzzification: At this step a set of predefined input variables $\{x_1, x_2, \dots, x_r\}$ are turned into linguistic forms by K_i fuzzy membership functions (MFs), i.e. $\{A_1^{(i)}, A_2^{(i)}, \dots, A_{K_i}^{(i)}\}$, where $i=1, \dots, r$. For our MFIS, triangular MFs are used for fuzzification procedure. Generally, to fuzzify any value, the minimum and maximum should be considered and based on the characteristics of inputs and the number of linguistic terms, the arrangement of MFs should be completed.

Step 2: Fuzzy rule base: Based on the knowledge of the expert and also the characteristics of the fuzzified inputs (number of MFs for each input, effectiveness of input etc.), a rulebase should be defined. Generally, a rule can be generated as:

IF inp_1 IS "ling inp_1 " ... AND/OR inp_r IS "ling inp_r " THEN out_1 IS "ling out_1 " AND... out_r IS "ling out_r "

Step 3: Fuzzy inference engine: In this paper, we use Mamdani's minimum operator. In other words, in the above symbol rule, AND operator is used to combine the MFs to calculate the firing rate of each rule in the rule base which in turn helps us calculating the MF values of the consequent part of each rule.

Step 4: Defuzzification: To calculate the total output of inference engine, center-of-sum (centroid) method is used. The mathematical definition of centroid method is given as:

$$MFIS_{out} = \frac{\int_w \sum_{i=1}^{ruleNum} w_i \mu_{Bi}(y) dy}{\int_w \sum_{i=1}^{ruleNum} \mu_{Bi}(y) dy} \quad (15)$$

where μ is the input of the defuzzifier and w is the aggregation quality. Once the defuzzification is completed, we have a surface which clearly indicates the effects of the input variation on the output values.

Fig. 6 indicates the overall scheme of the proposed expert system. As it can be seen, it enables us to define a map between the acquired data and the quality of robot maneuvers in laparoscopic surgery.

One of the important issues which should be taken into account is the computational complexity of the system after it is trained. This lets us evaluate the capabilities of a system for online applications. As it was mentioned, after training of the framework, the first phase (distributor) only determines the Euclidian distance of new observed pattern with the formed cluster centers to verify the proper ELM module for the new pattern. After that, the outputs of the activated module are fed to MFIS for linguistic inference. It is clear that the complexity of the first phases, i.e. computing the Euclidian distance of an observed pattern with the centers of clusters, can be easily neglected, as compared to the complexity of the processors of ELM and MFIS. The computational complexity of the second phase depends on the number of hidden nodes of the activated module (which varies based on the number of hidden nodes of the activated module). The complexity of the third phase depends on the number of linguistic rules taking part in the inference process. So, the total complexity of the sequential system depends on the number of hidden nodes of the activated ELM and the number of rules in MFIS.

3.3. Parameter settings and experimental setup

In this section, the operating and controlling parameters required for numerical experiments and algorithmic procedures are presented. As it was mentioned before, three types of algorithmic methods are used for (1) clustering, (2) identification and (3) fuzzy

interpretation. At clustering stage, several rival techniques are taken into account. The extracted methods are particle swarm optimization (PSO), accelerated chaotic particle swarm optimization (ACPSO), chaotic particle swarm optimization (CPSO), nelder-mead particle swarm optimization (NM-PSO), hybridization of K-Means and PSO (K-PSO) and K-Means [33,34]. All of these methods have been successfully used for classification task. In the original work on ACPSO, the mentioned algorithms, i.e. PSO, CPSO, NM-PSO, K-PSO and K-Means, were used as rival algorithms. Following a similar approach, in this paper, the authors would like to find out whether ACPSO is able to yield promising results for laparoscopic surgery. To conduct the clustering experiments, all rival methods execute the optimization for 5 independent runs. For clustering techniques, the authors decided to conduct it for 7, 8, 9, 10, 11, 12, 13 and 14 clusters (k). The input vector of data has 3 dimensions ($d=3$). As it was mentioned before, clustering is done for $10 \times d \times k$ iterations, and the number of particles is equal to $3 \times d \times k$. In each iteration, the inertia weight (w) in Eq. (13) is updated as:

$$\omega = 0.5 + \frac{rand(0, 1)}{2.0} \quad (16)$$

After termination of clustering experiments, the clustering results of rival methods are compared in terms of the following metrics [36,37]:

(a) **Compactness:** measures the within-cluster distance, and represents the compactness of clusters. For a problem with k clusters, compactness is calculated as:

$$Compactness = \frac{1}{k} \sum_{j=1}^k \frac{1}{n_j} \sum_{x_p \in c_j} Euc_dist(x_p, c_j) \quad (17)$$

Definitions of all parameters are as same as what stated in previous section. This metric should be minimized.

(b) **Separation:** measures the between-cluster distance, and indicates the separation of clusters. The separation metric is calculated as:

$$Separation = \frac{1}{k(k-1)} \sum_{i=1}^k \sum_{j=1}^k Euc_dist(c_i, c_j) \quad (18)$$

This metric should be maximized.

(c) **Ray-Turi's index:** is based on the ratio of intra-clustering distance to inter-clustering distance. The metric is calculated as:

$$Ray - Turi = (C \times \mathcal{N}(2, 1) + 1) \times \frac{intra}{inter} \quad (19)$$

where C is a user-specified parameter which is set to be 1, and $\mathcal{N}(\cdot)$ is a Gaussian distribution with $\mu=2$ and $\sigma=1$. The *intra* term is calculated using Eq. (17), and *inter* is obtained by:

$$inter = \min\{Euc_dist(c_i, c_j)\}, \quad i = 1, 2, \dots, l-1, \quad j = l+1, \dots, k \quad (20)$$

This metric should be minimized.

(d) **S_Dbw index:** is equal to the summation of average scattering of clusters and separation of clusters. The metric is mathematically expressed as:

$$S_Dbw = Scatter + Separation \quad (21)$$

where *Scatter* is obtained as:

$$Scatter = \frac{1}{k} \sum_{j=1}^k \frac{\|\psi(C_j, \forall x_p \in c_j)\|}{\|\psi(Y, \forall x_p \in set)\|} \quad (22)$$

where $\|\cdot\|$ is the Euclidian norm, and ψ indicates the variance of data involved in set/cluster j , and is defined as:

$$\psi(C_j, \forall x_p \in C_j) = \sqrt{\sum_{i=1}^d \left[\frac{1}{n_j} \sum_{n=1}^{n_j} (x_{n,i} - z_{j,i})^2 \right]^2} \quad (23)$$

where $z_{j,i}$ represents the i th dimension of the centroid of the j th cluster, and other definitions are the same as what presented previously.

The Separation measure is calculated as:

$$\text{Separation} = \frac{1}{k(k-1)} \sum_{i=1}^k \sum_{j=1, j \neq i}^k \frac{\varphi(m_{ij})}{\max\{\varphi(c_i), \varphi(c_j)\}}, \quad (24)$$

where m_{ij} is the middle point of cluster centers c_i and c_j , and is calculated as $m_{ij} = c_i + c_j / 2$, $\varphi(c_i)$ represents the density of

data points in the neighborhood of c_i , and is calculated as:

$$\varphi(c_i) = \sum_{n=1}^{n_i} h(c_i, x_n) \quad (25)$$

where

$$h(c_i, x_n) = \begin{cases} 1 & \text{if } Euc_dist(c_i, x_n) < \tilde{\psi} \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

where

$$\tilde{\psi} = \frac{1}{k} \sqrt{\sum_{i=1}^k \|\sigma(C_i)\|} \quad (27)$$

This metric should be maximized.

Table 1
Comparison of the fitness values for the rival clustering techniques.

	ACPSO	CPSO	NM-PSO	K-PSO	PSO	K-Means
Mean						
7 Clusters	596.013	666.832	809.923	813.327	846.028	845.545
8 Clusters	656.821	677.286	891.478	892.118	910.227	945.545
9 Clusters	694.554	728.132	1072.835	1194.726	979.619	1245.543
10 Clusters	726.032	883.679	981.399	881.089	1072.843	1308.556
11 Clusters	914.572	1007.68	1192.847	1060.067	1391.384	1364.948
12 Clusters	1020.078	999.893	1450.971	1283.132	1576.301	1375.795
13 Clusters	1049.042	1156.231	1305.994	1101.384	1795.942	1389.993
14 Clusters	1424.099	1273.889	1345.372	1720.622	1911.766	1398.384
Best						
7 Clusters	502.854	612.984	654.847	706.867	753.059	839.857
8 Clusters	608.992	628.248	710.938	784.857	812.478	932.847
9 Clusters	613.393	643.854	968.994	938.958	854.948	1109.736
10 Clusters	687.459	676.867	924.092	723.923	876.993	1298.877
11 Clusters	874.773	824.756	983.948	974.884	1024.985	1357.495
12 Clusters	978.711	985.837	1183.846	1085.962	1211.094	1368.039
13 Clusters	1004.955	1095.192	1284.049	1086.059	1342.767	1387.958
14 Clusters	1310.092	1269.857	1201.843	1495.039	1505.884	1394.069
Worst						
7 Clusters	810.948	842.948	1045.948	932.847	1043.986	861.764
8 Clusters	835.417	851.454	1227.847	942.334	1394.094	948.479
9 Clusters	892.099	912.802	1209.747	1298.647	1404.912	1284.756
10 Clusters	935.203	1232.811	1124.461	1313.067	1465.155	1362.948
11 Clusters	983.123	1391.183	1483.911	1373.847	1672.994	1387.576
12 Clusters	1294.948	1303.645	1493.092	1421.948	1776.710	1391.885
13 Clusters	1383.721	1342.847	1684.094	1434.002	2023.847	1423.857
14 Clusters	1674.857	1712.746	1723.948	2073.950	2437.811	1445.867
Std.						
7 Clusters	113.918	87.929	140.040	79.976	106.801	8.264
8 Clusters	87.698	86.034	187.671	57.826	231.227	6.139
9 Clusters	103.564	98.982	85.599	133.795	211.153	67.296
10 Clusters	99.271	200.383	58.788	221.4734	214.794	25.750
11 Clusters	39.108	207.659	178.190	154.323	230.279	11.418
12 Clusters	128.802	137.548	125.503	119.859	205.005	8.734
13 Clusters	155.776	94.071	171.808	150.611	248.575	15.397
14 Clusters	134.291	196.721	195.493	207.6443	331.125	21.797

Table 2
Computational time for each metaheuristic clustering technique.

	ACPSO	CPSO	NM-PSO	K-PSO	PSO
7 Clusters	51.723913	49.604856	116.917927	50.900509	49.344908
8 Clusters	75.070337	72.938164	201.719481	77.229579	74.896279
9 Clusters	112.601653	108.095736	435.815303	114.662101	111.576077
10 Clusters	167.495695	159.978476	659.603573	168.957362	164.739121
11 Clusters	234.636669	222.877751	873.927382	231.280659	229.601741
12 Clusters	307.387957	292.968374	1021.827333	306.992480	304.129973
13 Clusters	344.697298	329.094757	1108.948321	348.792617	336.219171
14 Clusters	449.847567	436.978129	1265.980374	450.485764	442.492145

- (e) **Dunn's index:** is used to find out how much clusters are well separated and compact. It is mathematically expressed as:

$$Dunn = \min_{1 \leq i \leq h} \left\{ \min_{h+1 \leq j \leq k} \left(\frac{\alpha(\forall x_p \in c_i; \forall x_q \in c_j)}{\beta(\forall x_p, x_q \in c_g)} \right) \right\} \quad (28)$$

where

$$\alpha(\forall x_p \in c_i; \forall x_q \in c_j) = \min(Euc_dist(x_p, x_q)) \quad (29)$$

$$\beta(\forall x_p, x_q \in c_g) = \max(Euc_dist(x_p, x_q)) \quad (30)$$

This metric should be maximized.

- (f) **DB index:** measures the average similarity between each cluster. It is mathematically expressed as:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j=1, \dots, k, i \neq j} \left\{ \frac{1/2\vartheta(\forall x_p \in c_i) + 1/2\vartheta(\forall x_p \in c_j)}{Euc_dist(c_i, c_j)} \right\} \quad (31)$$

where ϑ is the centroid diameter and is defined as:

$$\vartheta(\forall x_p \in c_j) = 2 \left[\frac{\sum Euc_dist(x_p, c_j)}{n_j} \right] \quad (32)$$

The above metrics are extensively used for comparative studies. For identification phase, together with M-ELM neural network (M-ELMNN), back propagation neural network (BPNN) [29], Modular

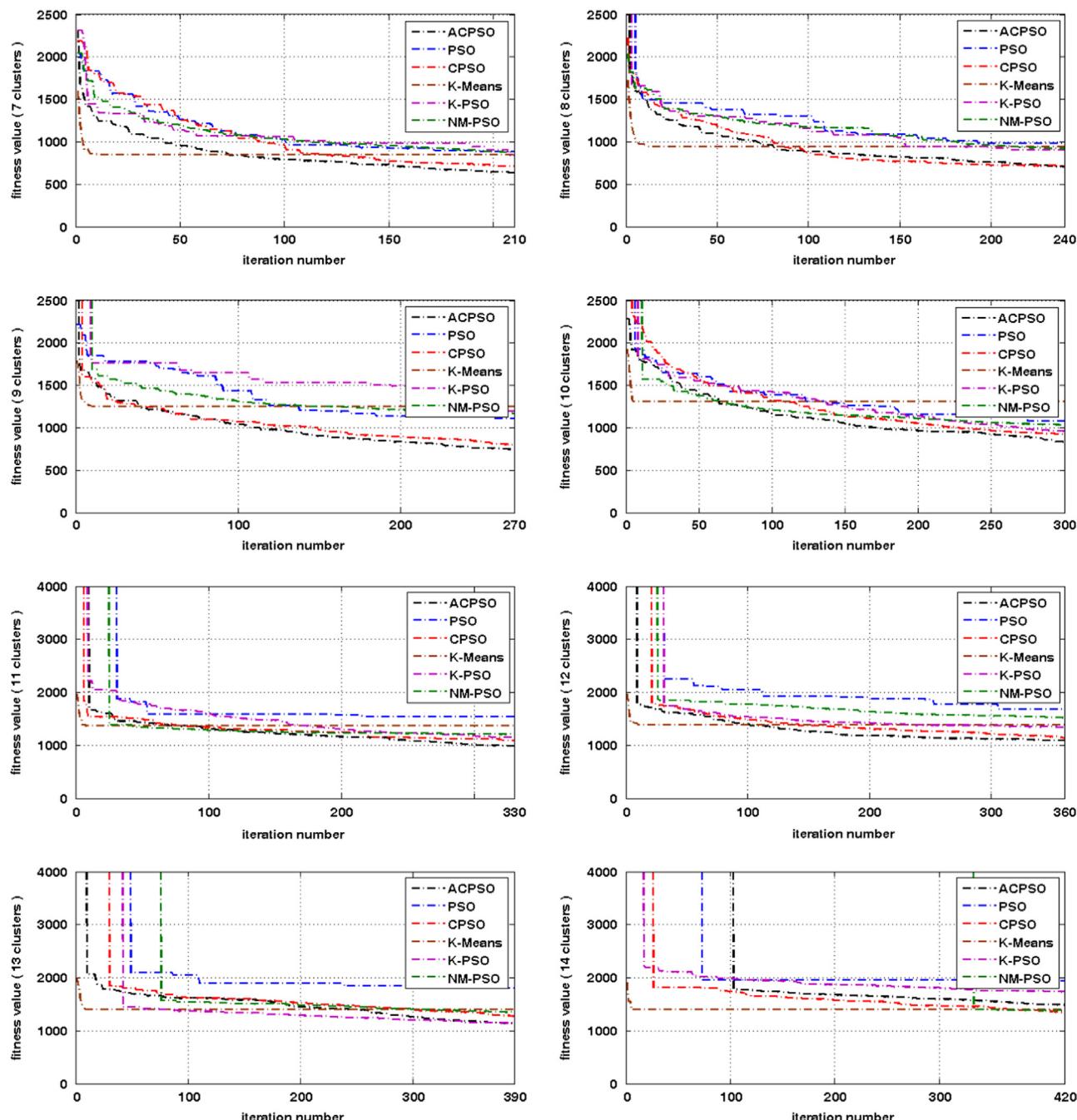


Fig. 7. Mean performance of the clustering techniques.

Table 3

Details of the cluster centers obtained by ACPSO (un-normalized).

Clusters	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	11th	12nd	13rd	14th
7 Cluster														
<i>inp</i> ₁	0.0126	0.0121	0.0126	0.0130	0.0120	0.0120	0.0124	–	–	–	–	–	–	–
<i>inp</i> ₂	0.3273	0.3671	0.3655	0.3327	0.3834	0.3733	0.3420	–	–	–	–	–	–	–
<i>inp</i> ₃	0.8329	0.7569	0.7114	0.7558	0.7246	0.7413	0.8023	–	–	–	–	–	–	–
8 Cluster														
<i>inp</i> ₁	0.0160	0.0157	0.0160	0.0161	0.0152	0.0155	0.0161	0.0161	–	–	–	–	–	–
<i>inp</i> ₂	0.4195	0.4389	0.3630	0.4084	0.4190	0.4198	0.3647	0.3908	–	–	–	–	–	–
<i>inp</i> ₃	0.8788	0.8241	0.8340	0.8621	0.7885	0.8943	0.8935	0.8052	–	–	–	–	–	–
9 Cluster														
<i>inp</i> ₁	0.0165	0.0173	0.0160	0.0155	0.0166	0.0174	0.0154	0.0151	0.0167	–	–	–	–	–
<i>inp</i> ₂	0.4688	0.3779	0.4583	0.4294	0.4148	0.3950	0.3835	0.4057	0.3410	–	–	–	–	–
<i>inp</i> ₃	0.8501	0.8617	0.8847	0.8177	0.7778	0.8008	0.9298	0.9166	0.9291	–	–	–	–	–
10 Cluster														
<i>inp</i> ₁	0.0121	0.0156	0.0140	0.0161	0.0123	0.0131	0.0150	0.0138	0.0159	0.0134	–	–	–	–
<i>inp</i> ₂	0.4403	0.3305	0.3673	0.2450	0.4724	0.3144	0.3584	0.3663	0.2265	0.3830	–	–	–	–
<i>inp</i> ₃	0.9171	0.6942	0.9843	0.8515	0.8008	1.1673	0.8238	0.8049	0.9472	0.9613	–	–	–	–
11 Cluster														
<i>inp</i> ₁	0.0168	0.0153	0.0213	0.0151	0.0157	0.0157	0.0185	0.0153	0.0198	0.0166	0.0198	–	–	–
<i>inp</i> ₂	0.3928	0.5526	0.4112	0.7193	0.4247	0.4506	0.2991	0.4076	0.4814	0.6852	0.3206	–	–	–
<i>inp</i> ₃	1.3828	0.7182	0.8664	0.9159	1.5642	1.3021	1.2404	1.1337	1.0998	0.8681	1.0944	–	–	–
12 Cluster														
<i>inp</i> ₁	0.0090	0.0144	0.0104	0.0203	0.0096	0.0177	0.0163	0.0183	0.0132	0.0105	0.0125	0.0125	–	–
<i>inp</i> ₂	0.4710	0.7096	0.3937	0.4608	0.3798	0.1080	0.3023	0.0098	0.2194	0.6889	0.7249	0.3115	–	–
<i>inp</i> ₃	1.3596	0.5757	0.9531	0.6475	1.5664	1.2060	0.3923	1.0125	1.4207	0.7791	0.7044	0.7498	–	–
13 Cluster														
<i>inp</i> ₁	0.0141	0.0148	0.0231	0.0154	0.0180	0.0226	0.0122	0.0164	0.0174	0.0164	0.0222	0.0112	0.0201	–
<i>inp</i> ₂	0.8229	0.8218	0.4323	0.7612	0.4278	0.5655	0.5225	0.4100	0.5688	0.4575	0.3778	0.6245	0.2728	–
<i>inp</i> ₃	0.6924	0.8741	1.0734	1.0107	1.2931	0.8647	0.9873	1.5817	0.6032	1.1269	0.8968	1.1627	1.4012	–
14 Cluster														
<i>inp</i> ₁	0.0234	0.0292	0.0214	0.0147	0.0190	0.0155	0.0169	0.0140	0.0316	0.0180	0.0151	0.0077	0.0181	0.0158
<i>inp</i> ₂	0.4806	0.4195	0.6528	0.4226	0.5708	0.5213	0.6089	0.5774	0.3929	0.8852	0.4098	0.6361	0.3243	0.7669
<i>inp</i> ₃	0.8640	0.7845	0.8097	1.0221	0.7073	1.4473	1.1836	2.2989	0.8262	0.8012	2.3892	1.4823	2.3521	1.0648

BPNN (M-BPNN), adaptive neuro-fuzzy inference system (ANFIS) [38], and ELMNN are used as rival techniques. For comparative study, testing and training errors as well as training time are considered as performance evaluation metrics. Mean squared error (*MSE*) is used as error measure metric. *MSE* is mathematically expressed as:

$$MSE = \frac{1}{N} \sum_{j=1}^N |(t_j - o_j)| \quad (33)$$

To find the optimum structure for all of those rival identifiers, several experiments are performed and the final architectures are extracted through trial and error procedure. For the sake of simplicity, and also avoiding any miss-interpretation, all of the data in the database are normalized spanning unity [0, 1] using the following map:

$$inp_i^{[0,1]} = \frac{inp_i - inp_i^{\min}}{inp_i^{\max} - inp_i^{\min}}, \quad i = 1, 2, 3 \quad (34)$$

where $inp^{\min} = \{0.0050, 0.0098, 0.0062\}$, $inp^{\max} = \{0.0396, 1.3076, 3.0210\}$, and $inp_i^{[0,1]}$ represents the normalized value of inp_i .

In this case study, *Logsig* activation function is considered for all neural networks. *Logsig* activation function is a very simple and computationally efficient mapping scheme which has clearly demonstrated its capabilities for machine learning and modeling tasks. In our previous studies, it is observed that *logsig* activation function can afford very promising incomes for ELM which is in a very good agreement with the reports available in the literature [40]. Moreover,

for verifying the algorithmic parameters and operating features of the proposed method, the authors conducted a comparative study and observed that *logsig* is a very good activation function for capturing the required knowledge from the collected database. For ANFIS topology, *Gaussian* MFs are selected through a trial and error procedure. For the iterative back-propagation-based soft identifiers, BPNN, M-BPNN and ANFIS are trained for 100 epochs which is verified through experiments.

After termination of the identification stage, the outputs of M-ELMNN are un-normalized to be prepared for fuzzy interpretation using the following equation:

$$inp_i = inp_i^{\min} + inp_i^{[0,1]} \times (inp_i^{\max} - inp_i^{\min}), \quad i = 1, 2, 3 \quad (35)$$

The detailed definitions of all steps of MFIS were provided in previous section. In the next section, the results of the experiments are presented based on the performance evaluation metrics stated in this section. Both implementations of the algorithms and numerical experiments are conducted on a PC with a Pentium IV, Intel Dual core 2.2. GHz and 1 GB RAM.

4. Results and discussion

Table 1 summarizes the statistical results of intra-cluster distances for all rival clustering techniques obtained after 5 independent runs. The better results are shown in bold. As it can be seen, in most cases,

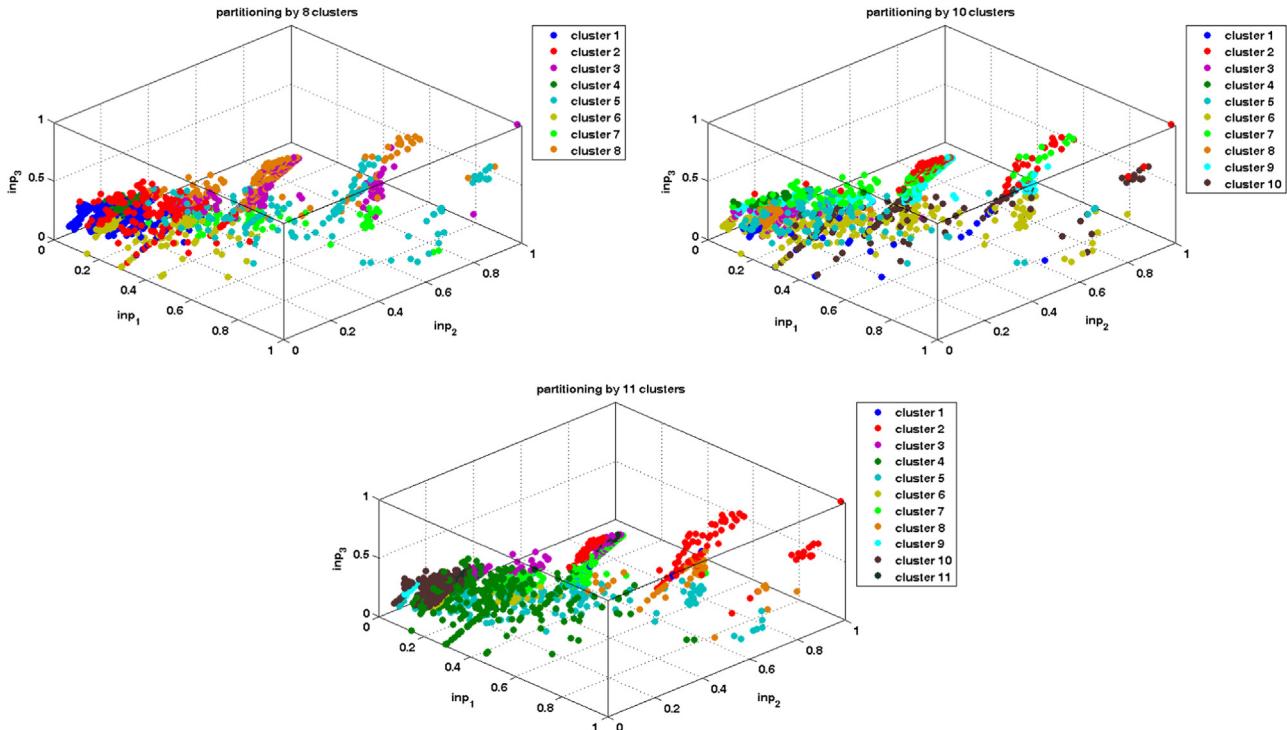


Fig. 8. Final clustering results for 8, 10 and 11 clustering scenarios.

Table 4

Comparison of the performance measures for each partitioning schemes (normalized data).

	7 Cluster	8 Cluster	9 Cluster	10 Cluster	11 Cluster	12 Cluster	13 Cluster	14 Cluster
Compactness	0.3540	0.3152	0.3595	0.3604	0.4641	0.5251	0.5120	0.6609
Separation	0.0340	0.0431	0.0659	0.1599	0.2260	0.3750	0.3217	0.4497
Ray-Turi's index	8.1979	6.0200	5.8660	5.6622	7.1874	2.5838	2.0958	2.1477
S_Dbw index	0.8761	0.6054	0.6516	0.7731	0.7981	0.6712	0.73019	0.8723
Dunn's index	0.0792	0.0531	0.0171	0.0386	0.0069	0.0256	0.0193	0.0235
DB index	47.9497	24.6021	25.3835	16.0313	11.8492	9.4587	8.1411	9.0352

i.e. 7, 8, 9, 10, 11 and 13 clusters, ACPSO outperforms the rival clustering techniques. For 12 and 14 clustering cases, CPSO yields better results. As it can be seen, among the rival techniques, the best performance belongs to algorithms which take the advantage of chaotic exploration/ exploitation. Similar results are achieved in terms of best obtained values. The only difference is in clustering with 14 clusters where the best result is obtained by NM-PSO. By taking a peek at results obtained in worst and std. stages, it can be interpreted that the chaotic clustering approaches are robust. As stochastic clustering techniques, such a promising result helps the authors to conclude that both ACPSO and CPSO can be reliably used for clustering problems. It is worth mentioning that although K-Means outperforms other techniques in terms of robustness, it suffers from several drawbacks such as trapping into local optimal solutions. This can be seen by tracking its performance for all clustering cases. As it can be seen, the obtained intra cluster distance of K-Means technique is around 1300–1400 for most cases. This suggests a better performance for cases with high number of clusters, and weaker performance for problems with lower number of clusters. NM-PSO and K-PSO compete with each other in terms of mean, best and std. results. As it can be seen, for some problems, NM-PSO outperforms the K-PSO technique while for some other cases K-PSO is the better one.

Table 2 lists the mean computational time of rival methods in 5 independent runs. The results help us making a more firm conclusion regarding the performance of NM-PSO and K-PSO. As it can be seen, the computational complexity of K-PSO is much lower

than NM-PSO. Therefore, although NM-PSO is a bit better than K-PSO, we can claim K-PSO is a more practical clustering approach for our problem. The other interesting observation is that CPSO is a bit faster than PSO algorithm. This is because, at each iteration, instead of random sampling, the random walk coefficients are calculated using a simple deterministic chaotic technique. As a general remark, it can be concluded that embedding chaotic behavior not only improves the performance of PSO, but also does not augment the computational complexity. This is while hybridizing nelder-mead and PSO increases the computational complexity significantly as for problems of higher clusters the computational complexity is approximately three times higher than other methods.

Fig. 7 compares the mean performance of the clustering techniques. From the first glance, one can easily observe that K-Means technique converges to a local optimal solution at the very beginning of the procedure. Besides, in all cases, except $k=14$, ACPSO indicates better results. One important observation is that neither of the chaotic-based clustering approaches show a bold premature convergence, and a gradual and permanent improvement of the performance can be observed for them. Among the swarm-based classifiers, PSO has a weaker performance which was predictable. This also implies that by using PSO and logistic chaotic map, we successfully increase the diversity of search. This is while the chaos does not hinder the beneficial exploitation of PSO.

Based on the results stated above, ACPSO is selected as the best clustering approach, and thus we analyze its performance to find the optimum partitioning scheme among the considered possibilities.

Table 5
Number of data handled by each model for different scenarios.

	7 Modules	8 Modules	9 Modules	10 Modules	11 Modules	12 Modules	13 Modules	14 Modules
Module 1	661	722	627	35	6	30	183	406
Module 2	59	174	167	97	141	208	227	14
Module 3	143	250	387	172	100	30	150	26
Module 4	354	283	24	51	372	28	164	13
Module 5	246	152	172	93	317	186	22	4
Module 6	370	173	135	614	176	36	305	32
Module 7	282	99	151	212	330	10	121	93
Module 8	0	262	135	126	61	626	36	189
Module 9	0	0	317	531	41	246	170	780
Module 10	0	0	0	184	318	20	356	18
Module 11	0	0	0	0	253	87	283	259
Module 12	0	0	0	0	0	608	80	158
Module 13	0	0	0	0	0	0	18	105
Module 14	0	0	0	0	0	0	0	18

Table 6
Number of hidden nodes of the ELM network for each module.

	7 Clusters	8 Clusters	9 Clusters	10 Clusters	11 Clusters	12 Clusters	13 Clusters	14 Clusters
Module 1	53	58	50	28	6	10	37	41
Module 2	30	52	33	29	28	62	45	13
Module 3	43	25	39	34	50	21	45	23
Module 4	35	57	24	31	74	20	33	12
Module 5	49	30	17	56	63	56	20	3
Module 6	30	35	14	49	35	32	31	22
Module 7	28	40	15	42	33	10	36	37
Module 8	0	52	27	25	36	50	22	57
Module 9	0	0	32	53	28	49	34	62
Module 10	0	0	0	55	31	20	36	16
Module 11	0	0	0	0	20	26	28	21
Module 12	0	0	0	0	0	51	32	32
Module 13	0	0	0	0	0	0	18	32
Module 14	0	0	0	0	0	0	0	14
Total nodes	268	349	251	402	404	407	417	385

Table 7

Training and testing error of each module and the overall identifier for tool-tissue force.

	7 Clusters	8 Clusters	9 Clusters	10 Clusters	11 Clusters	12 Clusters	13 Clusters	14 Clusters
Training error								
Module 1	0.0681	0.0646	0.0661	0.0156	0.1473e-8	0.2887e-14	0.0999	0.0496
Module 2	0.0133	0.0666	0.0923	0.0856	0.0545	0.0656	0.0554	0.0196
Module 3	0.0673	0.0964	0.0436	0.0962	0.1486	0.0371	0.0938	0.0552
Module 4	0.0757	0.1992	0.7685e-3	0.0565	0.0905	0.0330	0.1493	0.0009
Module 5	0.1108	0.0869	0.0777	0.0523	0.0892	0.0808	0.0007	0.4278e-10
Module 6	0.0752	0.0497	0.0825	0.0627	0.1101	0.0701	0.1056	0.0543
Module 7	0.0408	0.1225	0.0804	0.0823	0.1155	0.0603e-6	0.0217	0.0432
Module 8	-	0.1083	0.0915	0.0482	0.0204	0.0862	0.0760	0.0619
Module 9	-	-	0.1104	0.0588	0.0980	0.0577	0.0827	0.1023
Module 10	-	-	-	0.1150	0.1190	0.3496e-5	0.0611	0.0297
Module 11	-	-	-	-	0.1025	0.0586	0.0555	0.1328
Module 12	-	-	-	-	-	0.0615	0.0993	0.0827
Module 13	-	-	-	-	-	-	0.0194e-3	0.0614
Module 14	-	-	-	-	-	-	-	0.0368
Total	0.0704	0.0950	0.0746	0.0698	0.0998	0.0680	0.0787	0.0824
Testing error								
Module 1	0.0937	0.1045	0.0988	0.0775	0.9846e-6	0.8893e-8	0.1720	0.0892
Module 2	0.0663	0.0908	0.1009	0.1093	0.0875	0.1093	0.1036	0.0873
Module 3	0.1204	0.1209	0.0804	0.1443	0.2673	0.0984	0.1008	0.1982
Module 4	0.2056	0.2731	0.0432	0.0984	0.0367	0.1682	0.1827	0.0284
Module 5	0.2725	0.1635	0.1985	0.1093	0.0986	0.1345	0.0023	0.2854e-9
Module 6	0.1908	0.0809	0.2093	0.1209	0.2652	0.1483	0.1932	0.1093
Module 7	0.2018	0.1623	0.1809	0.1218	0.1012	0.1938e-5	0.1532	0.1923
Module 8	-	0.1663	0.2311	0.0888	0.1903	0.1923	0.1087	0.0893
Module 9	-	-	0.2212	0.1298	0.1204	0.1142	0.2932	0.2983
Module 10	-	-	-	0.1293	0.2093	0.3556e-4	0.1034	0.0884
Module 11	-	-	-	-	0.1108	0.2945	0.1806	0.1843
Module 12	-	-	-	-	-	0.1011	0.2021	0.1884
Module 13	-	-	-	-	-	-	0.5882e-3	0.0903
Module 14	-	-	-	-	-	-	-	0.0873
Total	0.1657	0.1405	0.1428	0.1216	0.1301	0.1401	0.1585	0.1911

Table 8

Training and testing error of each module and the overall identifier for maximum local stress.

	7 Clusters	8 Clusters	9 Clusters	10 Clusters	11 Clusters	12 Clusters	13 Clusters	14 Clusters
Training error								
Module 1	0.0510	0.0640	0.0765	0.0187	0.1046e-8	0.2887e-14	0.1050	0.0613
Module 2	0.0635	0.0821	0.0687	0.0527	0.0656	0.0761	0.0723	0.0020
Module 3	0.0563	0.0571	0.0444	0.1125	0.0930	0.0297	0.1168	0.0996
Module 4	0.0844	0.1136	0.8709e-3	0.0288	0.0860	0.0180	0.1071	0.0126
Module 5	0.0981	0.1068	0.0676	0.0683	0.0911	0.0454	0.0080	0.3064e-10
Module 6	0.0793	0.0677	0.1083	0.0381	0.0705	0.0543	0.0615	0.0676
Module 7	0.0505	0.0760	0.0661	0.1046	0.0988	0.1338e-6	0.0353	0.0388
Module 8	-	0.0956	0.1093	0.0856	0.0216	0.0696	0.1210	0.0696
Module 9	-	-	0.1212	0.0603	0.0739	0.0534	0.1289	0.0825
Module 10	-	-	-	0.0993	0.0516	0.0783e-5	0.0529	0.0228
Module 11	-	-	-	-	0.0758	0.0927	0.0543	0.1152
Module 12	-	-	-	-	-	0.0702	0.0691	0.0555
Module 13	-	-	-	-	-	-	0.1132e-3	0.0823
Module 14	-	-	-	-	-	-	-	0.0342
Total	0.0677	0.0792	0.0785	0.0660	0.0777	0.0639	0.0756	0.0753
Testing error								
Module 1	0.2091	0.0991	0.2021	0.0742	0.2443e-7	0.8371e-9	0.2192	0.0994
Module 2	0.1092	0.0908	0.1088	0.0293	0.1201	0.1211	0.1983	0.0001
Module 3	0.0901	0.0834	0.1035	0.1742	0.1094	0.0857	0.2093	0.1983
Module 4	0.2091	0.1882	0.0245	0.0812	0.2315	0.1878	0.1992	0.0884
Module 5	0.2341	0.1083	0.1291	0.1003	0.1211	0.2945	0.1845	0.98473-7
Module 6	0.1205	0.0982	0.1953	0.0892	0.1508	0.1985	0.1284	0.1821
Module 7	0.0902	0.1833	0.1009	0.2036	0.1511	0.2773e-9	0.1092	0.1434
Module 8	-	0.1093	0.2045	0.1912	0.1208	0.1092	0.2837	0.1153
Module 9	-	-	0.2145	0.0992	0.2298	0.1192	0.1936	0.1923
Module 10	-	-	-	0.2015	0.1011	0.0911e-6	0.0762	0.0809
Module 11	-	-	-	-	0.2904	0.1192	0.0978	0.1311
Module 12	-	-	-	-	-	0.1746	0.0882	0.0982
Module 13	-	-	-	-	-	-	0.1872e-2	0.1734
Module 14	-	-	-	-	-	-	-	0.1066
Total	0.1698	0.1143	0.1631	0.1232	0.1661	0.1462	0.1469	0.1445

Table 3 summarizes the un-normalized positions of cluster centers for all clustering scenarios obtained by ACPSO. Based on the position of the cluster centers, the data belonging to each cluster is verified. **Fig. 8** indicates the final clustering results for 8, 10 and 11 partitioning scenarios. Those results are important for extracting the optimum

partitioning scheme. Having the positions of cluster centroids for all partitioning schemes and data points enables us to calculate the values of the comparative metrics.

Table 4 depicts the performance measure values for all partitioning schemes. By analyzing the results, we came to the conclusion that

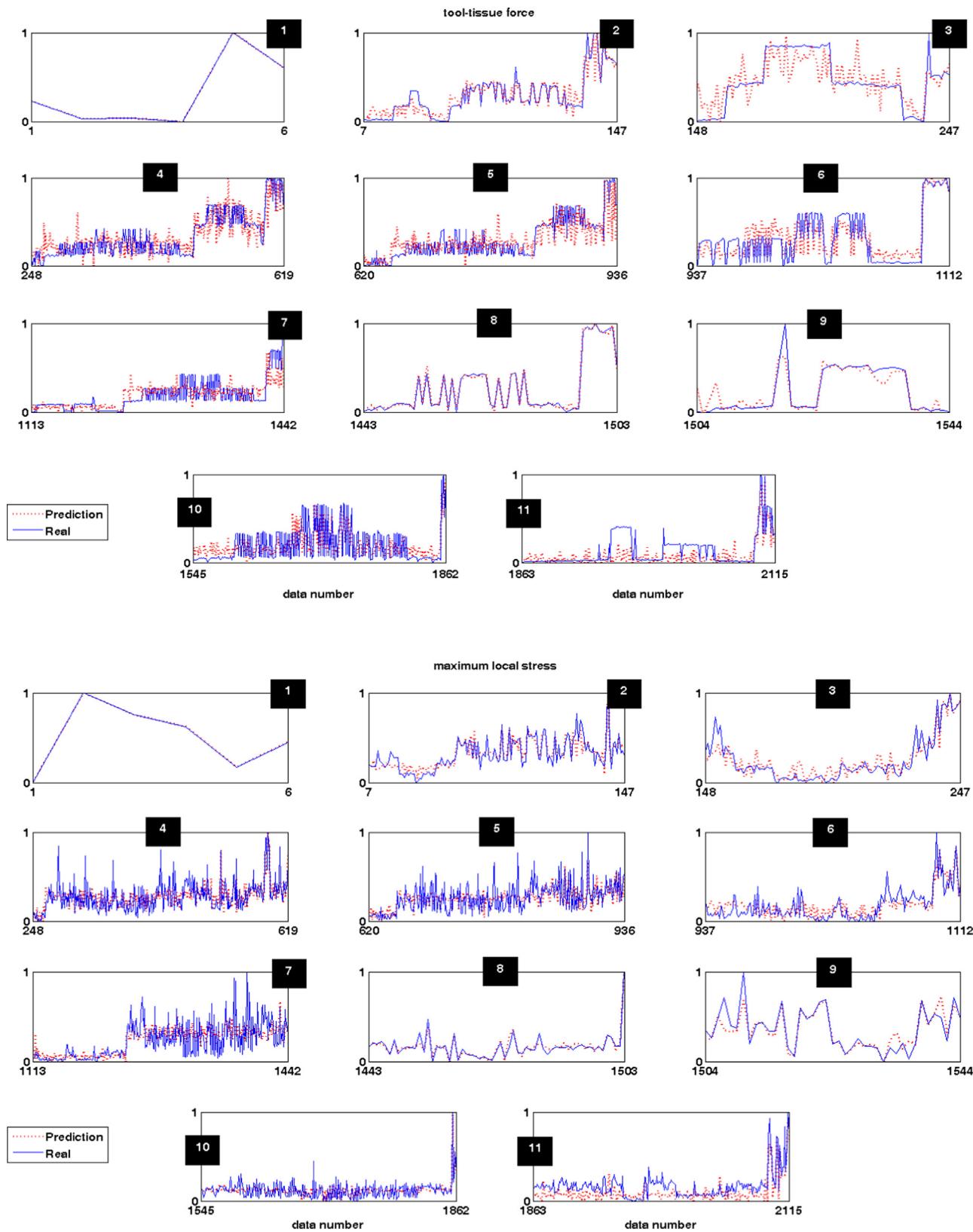


Fig. 9. Prediction accuracy of all modules for (a) tool-tissue force and (b) maximum local stress.

considering 10 clusters for laparoscopic data base yields acceptable results. However, such a conclusion cannot be drawn at this stage. For obtaining a more authentic result, we extend the comparative analysis to identification step. Obviously, for each clustering scenario, a consequent modular ELMNN is designed. **Tables 5 and 6** indicate the main characteristics of all modular identifiers, i.e. the number of data handled by each module, and number of hidden nodes in each ELMNN module. As it can be seen, in each scenario, each module of M-ELM has a specific number of hidden nodes. The main reason behind such a behavior lies in the fact that each of the clusters hosts specific types of patterns which result in knowledge-bases of different characteristics and nonlinearities. Therefore, capturing the underlying nonlinearity of the data of each cluster requires an ELM with specific computational capabilities (different number of hidden nodes). This implies that each of those modules should include proper number of nodes to correctly conduct the interpolation. Fortunately, the efficient algorithmic structure of ELM enables us to conveniently utilize an architecture selection mechanism to obtain the optimum number of hidden nodes for a given database. Among the existing methods, a very robust architecture selection mechanism, namely model order selection method [41], is used to verify the optimum number of hidden nodes. Based on the assumption that number of hidden nodes (computational units) has a direct impact on the complexity of modular system ($O(N)$), in **Table 6**, we calculate the total number of hidden nodes. As it can be seen, except modular identifiers with 7 and 9 modules, other networks have approximately equal computational complexity. For further investigation on the identification capabilities of the potential M-ELMNNs, in **Tables 7 and 8**, the training and testing MSE errors are tabulated for both tool-tissue force and maximum local stress. After calculating the MSE error of

each module, the total testing and training MSE errors are calculated as:

$$MSE_{total} = \frac{\sum_{i=1}^k (N_i \times MSE_i)}{N_1 + N_2 + \dots + N_k} \quad (36)$$

where N_i represents the number of data handled by the i th module, and MSE_i is the mean squared error of i th module obtained from Eq. (33).

By analyzing the results, it can be seen that M-ELMNN with 10 modules can effectively balance the prediction capabilities for both tool-tissue and maximum local stress estimation. Although there may be other modular architectures which can outperform M-ELMNN with 10 modules for a specific case, there is no identifier which can balance the prediction capabilities better than it. Such a conclusion is also in agreement with the results of the clustering phase. As it was mentioned, M-ELMNN with 10 modules outperform the rival modular architectures in terms of the clustering metrics. It can be concluded the better performance of the distributor (which is proportional to the separation of data, compactness of data, intra and inter clustering distances) can afford better identification results. The reason to this phenomenon is clear. As it was mentioned, each module uses extreme learning machine strategy to train the ANN. As ELM is based on linear algebraic equations and each independent cluster is fed to an ELMNN, the knowledge of a more compact cluster can be extracted much easier. **Figs. 9 and 10** depict the prediction error and correlation of the ELMNN with 10 modules, respectively. Those figures help us to clearly evaluate the performance of each module as well as the overall output of the entire network. As it can be observed from **Fig. 9(a and b)**, the modules show acceptable prediction results. Besides, **Fig. 10** implies that the entire M-ELMNN is also capable to effectively combine the

Table 9
Prediction error of the rival identification systems.

	Tool-tissue force		Maximum local stress		Time (s)
	Training error	Testing error	Training error	Testing error	
ELMNN	0.0854	0.2332	0.0572	0.1847	0.109840
M-ELMNN	0.0698	0.1216	0.0660	0.1232	2.706612
BPNN	0.0645	1.1198	0.0898	0.1802	58.254772
M-BPNN	0.0534	0.0892	0.0763	0.1093	152.28471
ANFIS	0.0728	1.3135	0.0481	0.1538	38.993412

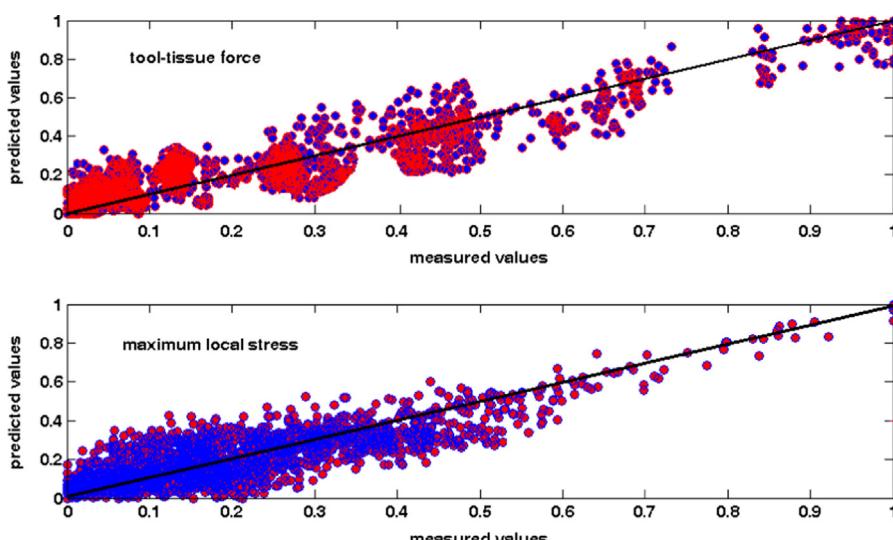


Fig. 10. Correlation between the output of the M-ELMNN and the experimental data.

results of its constructive modules, and consequently yield precise results. It is concluded that the ELM concept is quite capable of training the independent modules of the selected identifier.

After selecting the optimum architecture of the M-ELMNN, it is compared to the considered rival techniques. Table 9 summarizes the training and testing errors as well as computational complexity for rival identifiers. As it can be seen, the obtained MSE errors are quite comparable. It seems that the modular versions of both BP and ELM based ANNs act better than their sole versions. Besides, for our case study, the experiments suggest the use of ANN-based identifiers as compared to ANFIS (which is the integration of fuzzy and neuro computing concepts). From prediction point of view, it can be inferred that the use of gradient learning technique yields a bit better result as compared to the ELM-based ANNs. However, by further inspection, it can be seen that the training time of the BP-

based ANN is thousand times more than the one of the ELM-based identifiers. Therefore, the use of ELMNN and M-ELMNN is much more practical. The low training time of the ELM-based ANN facilitates its application to both control and real-time identification of robot maneuvers during laparoscopic surgery.

As it was mentioned before, once the important features of the robot maneuvers are predicted, the output signals are fed to the MFIS system for further process (post-process). The main goal is to convert the obtained numbers to a more interpretable form to help the surgeon evaluating the quality of the robot maneuvers. First of all, the MFIS inputs (M-ELMNN outputs) and output should be expressed by a set of appropriate linguistic expressions. *Quality of robot maneuver* is selected as the output of MFIS. Fig. 11 (a) indicates the considered MFs for the input and output of the MFIS. The center of the MFs and their distribution over the data-

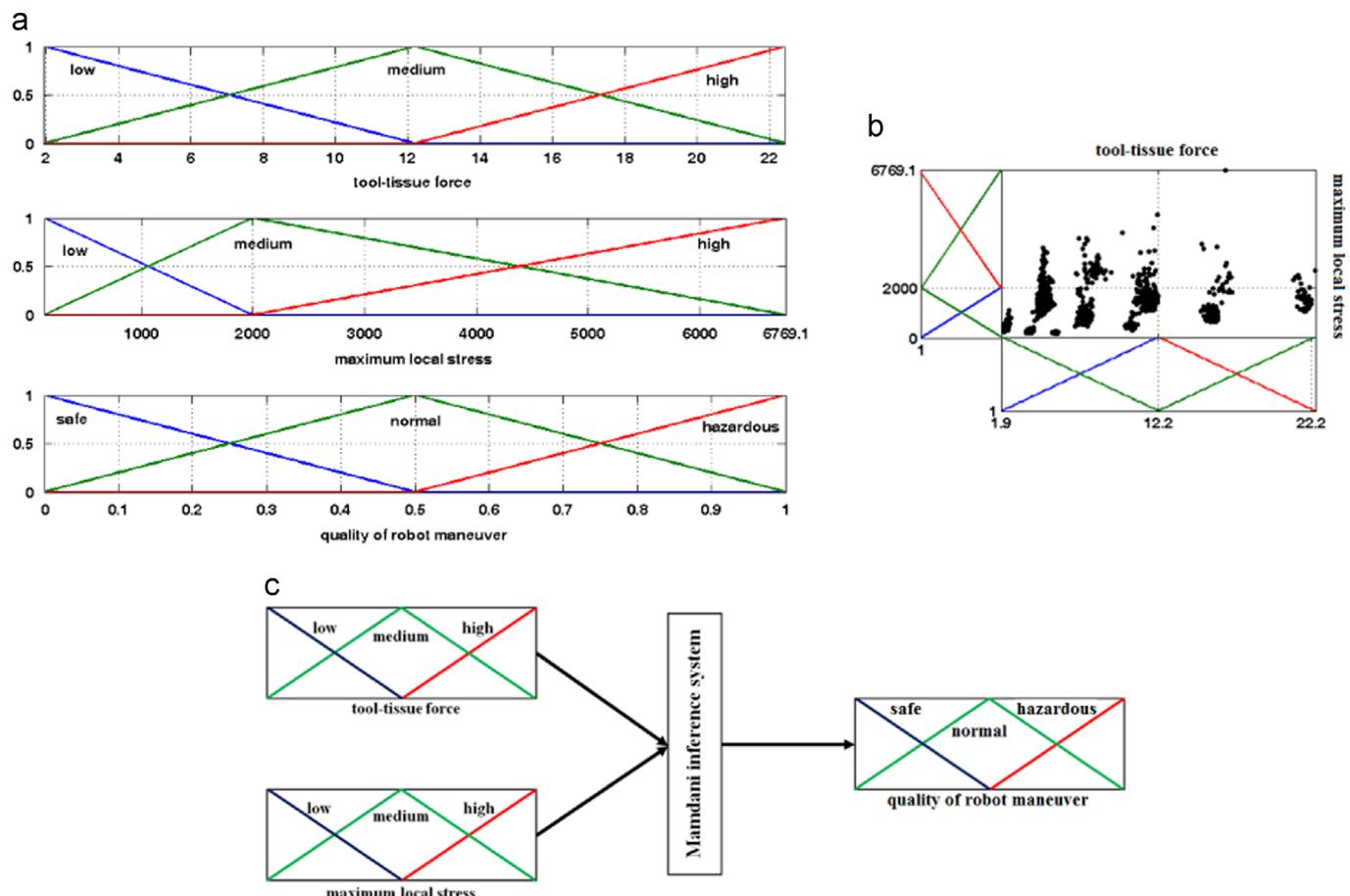


Fig. 11. (a) MFs of MFIS, (b) experimental data in partitioned space and (c) fuzzy inference system.

Table 10

The rules of the developed MIS.

Fuzzy controller/rule base

- 1: if tool-tissue force is low AND maximum local stress is low THEN quality of robot maneuver is safe
- 2: if tool-tissue force is low AND maximum local stress is medium THEN quality of robot maneuver is normal
- 3: if tool-tissue force is low AND maximum local stress is high THEN quality of robot maneuver is normal
- 4: if tool-tissue force is medium AND maximum local stress is low THEN quality of robot maneuver is safe
- 5: if tool-tissue force is medium AND maximum local stress is medium THEN quality of robot maneuver is normal
- 6: if tool-tissue force is medium AND maximum local stress is high THEN quality of robot maneuver is hazardous
- 7: if tool-tissue force is high AND maximum local stress is low THEN quality of robot maneuver is normal
- 8: if tool-tissue force is high AND maximum local stress is medium THEN quality of robot maneuver is normal
- 9: if tool-tissue force is high AND maximum local stress is high THEN quality of robot maneuver is hazardous

base are determined based on the experiments, knowledge of expert and sensitivity analysis. It seems that the allocated shape of the MFs is quite close to what used in practice. After providing the results, we can easily conduct the *Fuzzification* process. Fig. 11

(b) indicates the position of the MFIS inputs (obtained by M-ELMNN) in fuzzified space. This helps us to easily allocate them to proper linguistic MFs. As it is illustrated in Fig. 11(c), the fuzzified patterns are then transferred to the designed inference

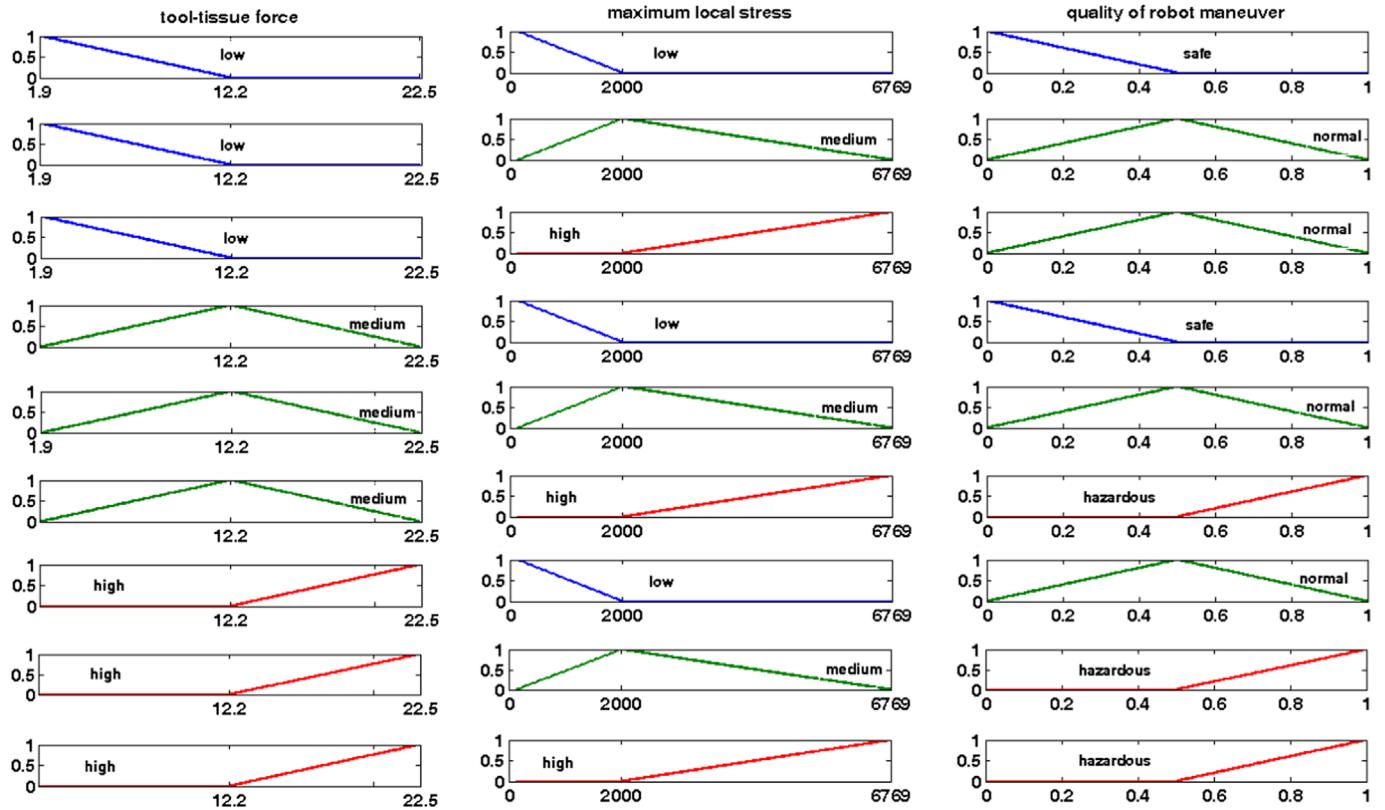


Fig. 12. Characteristics of the inference engine.

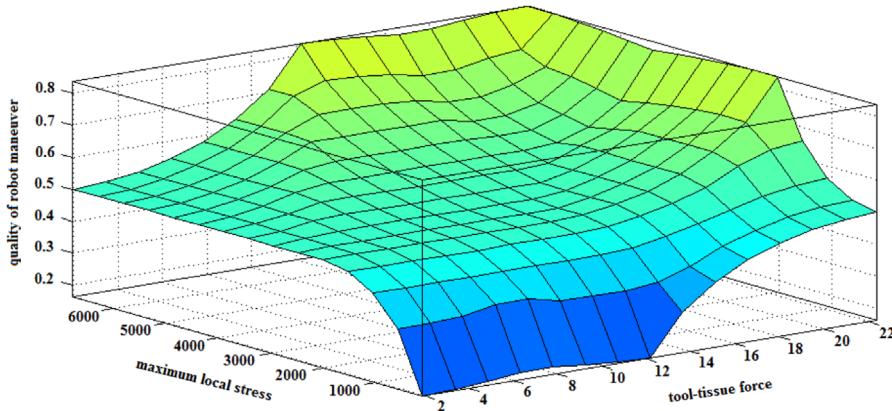


Fig. 13. The resulted surface of the MFIS.

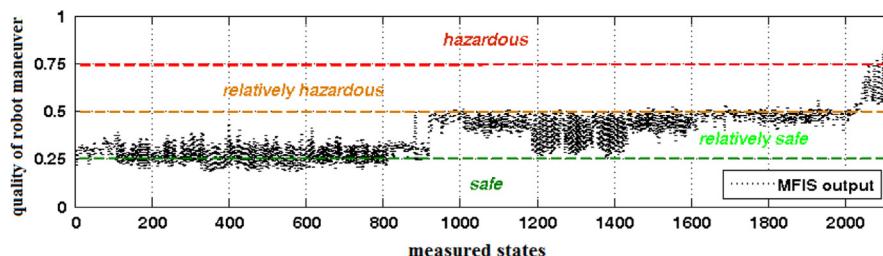


Fig. 14. Outputs of the MFIS for the experimental data.

engine to yield the quality of the robot maneuvers. The detailed descriptions of rules in the rulebase of the MFIS are presented in Table 10. Fig. 12 indicates the characteristics of the inference engine. After termination of the inference, a fuzzy-based map between the tool-tissue force, maximum local stress and quality of the robot maneuvers is derived as shown in Fig. 13. As it can be seen, the resulted surface is smooth and continuous which implies that the provided linguistic rules of the MFIS are capable of interpreting the quality of the robot motion at any condition. Fig. 14 shows the final output of the MFIS for all experimental results. As it can be seen, most of the measured geometrical features correspond to safe or relatively safe position of the surgery robot. However, the MFIS also covers the features which indicate hazardous movements, and therefore can be considered as a reliable interpreter for laparoscopic surgery operation.

5. Conclusions

In this paper, a hierarchical intelligent expert system was proposed to interpret the quality of the robot maneuvers during laparoscopic surgery. The method was constructed based on the integration of a swarm-based clustering approach, modular extreme learning machine (M-ELM) identifier and Mamdani fuzzy system. To endorse the authenticity of the method for practical application, rigor numerical and comparative experiments were performed. Based on the outcome of conducted experiments, the authors observed that the proposed expert system can easily cluster the geometrical features obtained from the 3D images, and then use them for the identification of the tool-tissue force and maximum local stress. The last phase of the proposed intelligent tool used the identification results to provide a more sensible output for the surgeon. The main advantages of using a fuzzy inference system at the last phase of the proposed expert tool are twofold. In one hand, by converting the numerical outputs of identifier to linguistic terms, it helps surgeons to easily evaluate the maneuvers of the surgery robot. On the other hand, the essence of fuzzy programming and linguistic rulebase effectively mitigates the effects of practical uncertainties. The authors' research also covered the use of M-ELM. Based on the experiments, it was observed that such a training policy can provide remarkable computational incomes. Fast and easy computation, mitigating the effects of un-desired complexities, and analytically supported decisions were found to be the most important characteristics of the M-ELM approach. Besides, it was observed that such a reliable, accurate and computationally efficient expert system can be easily used for incremental environments. In other words, the proposed identifier can be integrated into the robotic surgery system to incrementally learn and respond to real-time reactions of the surgeon during the laparoscopic surgery. Based on all of the above-mentioned traits, the authors conclude that the proposed expert system is useful and reliable for practical applications such as robotic laparoscopic surgery.

References

- [1] H. Dehghani-Ashkezari, A. Mirbagheri, F. Farahmand, S. Behzadipour, K. Firoozbakhsh, Real time simulation of grasping procedure of large internal organs during laparoscopic surgery, *IEEE Conf. Eng. Med. Biol. Soc.* (2012) 924–927. <http://dx.doi.org/10.1109/EMBC.2012.6346083>.
- [2] M. Tirehdast, A. Mirbagheri, M. Asghari, F. Farahmand, Modeling of interaction between a three-fingered surgical grasper and human spleen, *Stud. Health Technol. Inf.* 163 (2011) 663–669.
- [3] M. Hadavand, A. Mirbagheri, S. Behzadipour, F. Farahmand, A novel remote center of motion mechanism for the force-reflective master robot of haptic tele-surgery system, *Int. J. Med. Rob. Comput. Assisted Surg.* (2013), <http://dx.doi.org/10.1002/rcs.1515>.
- [4] M. Kohani, S. Behzadipour, F. Farahmand, Tool-tissue force estimation in laparoscopic surgery using geometric features, *Studies in Health Technology* and Informatics, Vol. 184, Medicine Meets Virtual Reality, IOS Press, Amsterdam, Netherlands (2012) 225–229.
- [5] E. Basafa, F. Farahmand, Real-time simulation of the nonlinear visco-elastic deformation of soft tissues, *Int. J. Comput. Assisted Radiol. Surg.* 6 (2011) 297–307.
- [6] K. Miller, Constitutive model of brain tissue suitable for finite element analysis of surgical procedures, *J. Biomech.* 32 (1999) 531–537.
- [7] H.W. Nienhuys, Van Der A.F. Stappen (2004) A computational technique for interactive needle insertions in 3D nonlinear material, in: *IEEE International Conference on Robotics and Automation Vol. 2004*, Orlando, pp. 2061–2067.
- [8] M. Sharifi, S. Behzadipour, G.R. Vossoughi (2012) Model reference adaptive impedance control of rehabilitation robots in operational space, in: *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, Rome, Italy, pp. 1698–1703.
- [9] C. Gholipour, M.B. Abolghasemi Fakhree, R. Alizadeh Shalchi, M. Abbasi, Prediction of conversion of laparoscopic cholecystectomy to open surgery with artificial neural networks, *BMC Surg.* (2009), <http://dx.doi.org/10.1186/1471-2482-9-13>.
- [10] M. Greminger, B.J. Nelson (2003) Modeling elastic objects with neural networks for vision-based force measurement, in: *IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, USA, pp. 1278–1283.
- [11] O.M. Becker, E. Araujo, JLMC Azevedo, Laparoscopy pneumoperitoneum fuzzy modeling, *Nat. Proc.* (2011), <http://dx.doi.org/10.1038/npre.2011.6060.1>.
- [12] B. Estebanez, P. Del Saz-Orozco, I. Rivas, E. Bazuano, V.F. Munoz, I. Garcia-Morales (2012) Maneuvers recognition in laparoscopic surgery: Artificial neural network and hidden Markov model approaches, in: *IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics*, Rome, Italy, pp. 1164–1169.
- [13] A. Mozaffari, S. Behzadipour, M. Kohani, Identifying the tool-tissue force in robotic laparoscopic surgery using neuro-evolutionary fuzzy systems and a synchronous self-learning hyper level supervisor, *Appl. Soft Comput.* 14 (2014) 12–30.
- [14] J. Huang, S. Payandeh, P. Doris, I. Hajshirmohammadi, Fuzzy classification: towards evaluating performance on a surgical simulator, *Stud. Health Technol. Inf.* 111 (2005) 194–200.
- [15] C. Feng, J.W. Rozenblit, A. Hamilton (2008) Fuzzy logic-based performance assessment in the virtual, assistive surgical trainer (VAST), in: *IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, Belfast, pp. 203–209.
- [16] G. Feng, G.B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning, *IEEE Trans. Neural Networks* 20 (2009) 1352–1357.
- [17] G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70 (2006) 489–501.
- [18] G.B. Huang, Learning capability and storage capability of two-hidden-layer feedforward networks, *IEEE Trans. Neural Networks* 14 (2003) 274–281.
- [19] M. Kohani, Real-Time Vision-Based Approach for Estimating Contact Forces on Soft Tissues, with Applications to Laparoscopic Surgery (M.Sc. Dissertation), Sharif University of Technology, 2013.
- [20] K. Lister, Z. Gao, J.P. Desai, Development of in vivo constitutive models for liver: application to surgical simulation, *Ann. Biomed. Eng.* 39 (2011) 1060–1073.
- [21] G.B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey, *Int. J. Mach. Learn. Cyber.* 2 (2011) 107–122.
- [22] G.B. Huang, L. Chen, Convex incremental extreme learning machine, *Neurocomputing* 70 (2007) 3056–3062.
- [23] G.B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine, *Neurocomputing* 71 (2008) 3460–3468.
- [24] X.Z. Wang, Q.Y. Shao, Q. Niao, J.H. Zhai, Architecture selection for networks trained with extreme learning machine using localized generalization error model, *Neurocomputing* 102 (2013) 3–9.
- [25] S.B. Kapil, On extreme learning machine for ϵ -insensitive regression in the primal by Newton method, *Neural Comput. Appl.* 22 (2013) 559–567.
- [26] C.W.T. Yeo, M.H. Lim, G.B. Huang, A. Agrawal, Y.S. Ong, A new machine learning paradigm for terrain reconstruction, *IEEE Geosci. Remote Sens. Lett.* 3 (2006) 382–386.
- [27] D. Serre, *Matrices: Theory and Applications*, Springer, New York, NY, 2002.
- [28] Y. Ding, Q. Feng, T. Wang, X. Fu, A modular neural network architecture with concept, *Neurocomputing* (2013), <http://dx.doi.org/10.1016/j.neucom.2012.06.051>.
- [29] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer-Verlag, Berlin, 1996.
- [30] A. Farooq, *Biologically Inspired Modular Neural Networks* (Ph.D. Dissertation), Virginia Tech, 2000.
- [31] M. Javadi, S.A.A. Abbaszadeh Arani, A. Sajedin, R. Ebrahimpour, Classification of ECG arrhythmia by a modular neural network based on mixture of experts and negative correlated learning, *Biomed. Signal Process. Control* 8 (2013) 289–296.
- [32] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, A.C.P.L.F. Carvalho, A survey of evolutionary algorithms for clustering, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* 39 (2009) 133–157.
- [33] L.Y. Chuang, C.J. Hsiao, C.H. Yang, Chaotic particle swarm optimization for data clustering, *Expert Syst. Appl.* 38 (2011) 14555–14563.
- [34] Y.T. Kao, E. Zahara, I.W. Kao, A hybridized approach to data clustering, *Expert Syst. Appl.* 34 (2008) 1754–1762.
- [35] L.A. Zadeh, Is there a need for fuzzy logic? *Inf. Sci.* 178 (2008) 2751–2779.

- [36] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, *J. Intell. Inf. Syst.* 17 (2001) 107–145.
- [37] A. Ahmadi, F. Karray, M.S. Kamel, Model order selection for multiple cooperative swarms clustering using stability analysis, *Inf. Sci.* 182 (2012) 169–183.
- [38] A. Mozaffari, A. Fathi, A. Khajepour, E. Toyserkani, Optimal design of laser solid freeform fabrication system and real-time prediction of melt pool geometry using intelligent evolutionary algorithms, *Appl. Soft Comput.* 13 (2013) 1505–1519.
- [39] G.H. Golub, C.F. Van Loan, *Matrix Computations*, second ed., John Hopkins University Press, Baltimore, Maryland, 1989.
- [40] A. Mozaffari, N.L. Azad, Optimally pruned extreme learning machine with ensemble of regularization techniques and negative correlation penalty applied to automotive engine coldstart hydrocarbon emission identification, *Neurocomputing* 131 (2014) 143–156.
- [41] N.Y. Liang, G.B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feed-forward networks, *IEEE Trans. Neural Networks* 17 (2006) (2006) 1411–1423.



Saeed Behzadipour received his B.Sc. in Mechanical Engineering from Sharif University of Technology, Iran. He received his M.Sc. and Ph.D. in Mechanical Engineering from University of Waterloo, Canada. His research interests include intelligent systems, robotics and biomechanical systems. He is an associate professor with the Mechanical Engineering Department at Sharif University of Technology.



Ahmad Mozaffari is a member of Smart Hybrid and Electric Vehicle Systems research group, in department of Systems Design Engineering at University of Waterloo, and a member of Robotic Surgery research group, at Sharif University of Technology. His research interests lie in engineering optimization, system identification, model predictive control, computational intelligence, and bio-inspired computing. He is the author of 43 refereed journal papers and 5 conference papers. He holds the 2013 Emerald's highly commended award for his contribution to International Journal of Intelligent Computing and Cybernetics.