

Selection and impact of different topologies in multi-layered hierarchical fuzzy systems

Juliusz Zajaczkowski, Brijesh Verma

phone: +61 7 3896 9439

fax: +61 7 3896 9843

juliusz.zajaczkowski@climatechange.qld.gov.au, b.verma@cqu.edu.au

Abstract An evolutionary algorithm based approach for selection of topologies in hierarchical fuzzy systems (HFS) is presented. Coupling fuzzy system with evolutionary algorithm provides a solution to the automated acquisition of the fuzzy rule base. It is difficult to study the problem of hierarchical decomposition for a large class of fuzzy systems but it is possible to analyse such architectures on the example of a particular fuzzy system, such as inverted pendulum. Topology of the HFS must be selected according to the physical properties of the dynamical system under consideration. Different HFS topologies for an inverted pendulum system are investigated and analysed to address the problem of how input configuration in multi-layered structure affects the controller performance. The experiments are conducted to test controller performance for different topologies of the hierarchical fuzzy system. The impact of different topologies on control process is discussed. The results from the case study of inverted pendulum can be extended to other dynamical systems.

keywords: fuzzy logic, evolutionary algorithms, inverted pendulum

1 Introduction

The design of control systems for complex and high dimensional dynamical systems relies on the availability of a system model under consideration. It is often difficult to create an adequate model of the system or process due to a limited availability of mathematical theory in case of very complex systems. Approximate models are often employed in such cases but with the growing discrepancy between physical system and its mathematical (or experimental) model. However, very complex systems can be controlled by human operators with only a rudimentary knowledge of the dynamic model. This kind of control problems has given rise to new intelligent control methods, fuzzy logic and neural networks being most widely used.

There are two main problems with the design of the intelligent control methods. The first is to obtain an adequate knowledge base for the controller, usually obtained from expert knowledge, and the second problem is selection of key parameters defined in the method. Evolutionary algorithms are often used for automated knowledge acquisition for fuzzy logic controllers [1—3]. However, there are a number of methods employed to knowledge base acquisition [2]:

- Fuzzy rule base derived from human experts. The expert specifies the linguistic labels associated with linguistic variables, structure of the rule base, and the meaning of each label.
- Fuzzy rule base derived from automated learning methods.

The most popular application of fuzzy set theory are fuzzy rule-based systems as they provided the framework for engineering applications. There are three major types of rule-based systems [4]:

- Linguistic fuzzy model in which both the antecedent and consequent part of *IF-THEN* rule are fuzzy propositions [5].
- Fuzzy relational model in which a particular antecedent proposition can be associated with several different consequent propositions via a fuzzy relation [6].
- Takagi-Sugeno fuzzy model in which the consequent is a crisp function of antecedent variables [7].

In this paper we take classical Zadeh [5] approach to fuzzy control.

In a hierarchical fuzzy logic structure, typically the most influential parameters are chosen as the system variables in the first level, the next most important parameters are chosen as the system variables in the second level, and so on [8]. In this hierarchy, the first level gives an approximate output which is then modified by the second level rule set, this procedure can be repeated in succeeding levels of hierarchy.

In general, with n input variables and m fuzzy sets defined for each input variable, there is m^n fuzzy rules in the rule base. In the hierarchical structure, the number of rules in a complete rule set is so reduced to a linear function of the number of variables, but this number may still be high. Increasing the number of input variables or input fuzzy sets results in an exponential increase in complexity of the rule base. The decomposition of the system into hierarchical fuzzy system is intended to reduce the size of the rule base while maintaining an adequate accuracy. Layered fuzzy logic systems utilize the modularity characterising many physical systems and their mathematical models. The output influenced by one closely related group of input variables may be largely independent of the values of other variables. Therefore, a layered fuzzy logic system can decompose the rule base along these lines of weak interdependence and still maintain a high level of accuracy.

The combination of fuzzy logic control (FLC) and evolutionary algorithm (EA) provides efficient method to examine different control systems for a given control problem. The FLC approach is used to define the framework used by solution search method (EA). Determining a particular HFS, and its encoding method, creates a search space within which the EA searches for the best solution according to the pre-defined performance index, see Section 6 for details. In other words, the fuzzy logic defines the control problem and the EA is used to find a solution to that problem. Evolutionary algorithm is a search

technique that mimicks the biological evolutionary strategies. The search space is initially filled with an initial population of potential solution, either randomly generated or initialized by any other mechanism, and then the population is subjected to evolutionary operators, such as selection operator ('survival of the fittest'), crossover (playing role of sexual reproduction), mutation, etc. Next population (often called generation) is created and every individual in the population is assigned the fitness value according to pre-defined fitness function. Fitness value is used to select the best individuals (potential solutions) for crossover operator. The process continues until termination condition is satisfied. If the EA ends successfully the solution to the control problem is found.

In this paper (extended version of [9,10]) different topologies of HFS for a given dynamical system (the inverted pendulum system) are investigated and analysed to address the problem of how input configuration in multi-layered structure affects the controller performance. The research presented in this paper has originated in research work by R.J. Stonier *et al.* [11]. Examination of the topologies gives insight into the workings of the physical system and its control system. For the inverted pendulum system, a single layer, two layers, three layers, and four layers HFS with different input configurations are examined and controllers' performances compared. There are various approaches to building a topology of the HFS and it is area of current research, see the following section.

2 Related work

There is a vast literature on fuzzy control systems, especially with applications to the inverted pendulum (cart-pole system) as it is often used as a test system for proposed methods. However, there is much less publications on hierarchical fuzzy control systems and topology selection. A large number of control systems (especially from 1980's and 1990's) rely on local linearization of the dynamical system under consideration. Design of the stabilizing fuzzy logic controllers is achieved via piece-wise linearization of the non-linear system, especially when authors are implementing Lyapunov direct method. Lyapunov method can be used not just for stability analysis but also to design fuzzy controllers, for example [12—14].

It is difficult to study the problem of hierarchical decomposition for a large class of fuzzy systems but it is possible to analyse such architectures on the example of a particular fuzzy system. Obviously, topology of the HFS must be selected according to the physical properties of the dynamical system under consideration. The selection process is subject to human decision. It might be possible to design the EA for finding the most suitable (optimal or near-optimal) topology for any particular problem (hierarchical EA), so the process can be automated. There is a number of papers dealing with variable

control structures, see for example [15—19]. P. Durr and C. Mattiussi [15] utilise a modular network architecture for the neural network and introduce its modular genetic representation that allows evolutionary algorithm to search for the neural network topology. The methodology can be implemented in fuzzy logic control where neural network is replaced by hierarchical fuzzy structure. A different approach is taken by G. Acampora [16] who designs a fuzzy controller with a variable configuration (among other characteristics) based on the concept of Timed Automata-based Fuzzy Controllers. The timed fuzzy controller acts on the system through a number of time intervals (known as control eras) which are characterized by specific control configurations which in turn are determined by: the number and typology of fuzzy variables and also by the number and structure of relationships between variables [16]. The same concept is explored in [17] with more specific application to modeling variable fuzzy structures.

The curse of dimensionality remains an unsolved problem in fuzzy logic control theory [20]. The problem is subject of many research papers with some authors focusing on systematic design of fuzzy logic systems. An automatic design of Takagi-Sugeno hierarchical fuzzy systems is investigated in [21]; the algorithm starts with random structures and rules' parameters. First, hierarchical structure is modified to find controller improvement and after completion of this step the rules' parameters are fine-tuned. The algorithm continues until a satisfactory solution (hierarchical TS-FS model) is found or a time limit is reached. Takagi-Sugeno approach has advantage of making use of Lyapunow stability analysis.

A number of papers investigate construction of hierarchical fuzzy systems. Most notably a series of papers in the special issue of International Journal of Intelligent Systems on hierarchical fuzzy systems. A review of the construction of hierarchical fuzzy systems is given in V. Torra paper [22]. E. Tunstel *et al.* [23] examines hierarchical control design and synthesis for the collection of subsystems comprising of fuzzy logic controllers and fuzzy knowledge-based decision systems. The technique is implemented to hierarchical behavior-based controllers for autonomous navigation of mobile robots. L. Magdalena [24] in his article analyses the role of context in hierarchical fuzzy controllers based on the decomposition of the input space. His aim is to improve the HFS design process by making it easier to introduce the expert knowledge in that process. K. Tachibana and T. Furuhashi [25] use multi-objective genetic algorithm (MOGA) and human expert intervention to determine hierarchical structure of submodels, select input variables of each submodel, divide input and output space, tune membership functions, and decide on the inference engine method. MOGA is used for selecting input variables of submodels. MOGA finds multiple models with different input variables and different numbers of fuzzy

rules. A human expert decides on the selection of the most suitable model. H. Kikuchi *et al.* [26] investigate partitioning of complex systems into simpler subsystems. They examined problem of an arbitrary function being decomposed into several subfunctions with a non-disjoint partition of variables. This problem corresponds to hierarchical fuzzy system decomposition. Their investigation established functional incompleteness of joint expansion which imposes certain limitation on hierarchical modelling.

Ming-Ling Lee *et al.* [27] introduces a new mapping rule base scheme to generate the fuzzy rule-base for the HFS. O. Cordon *et al.* [28] analyses problems related to the inflexibility of the linguistic rule. An extension of the knowledge base of linguistic fuzzy rule-based systems is introduced: the hierarchical knowledge base. M.R. Delgado *et al.* [29] introduces a hierarchical evolutionary approach to optimize the parameters of Takagi–Sugeno fuzzy systems. A least-squares method is used to determine the parameters of nonlinear consequents. A pruning procedure is implemented to prevent redundancy in each rule consequent. Their approach provides structurally simple fuzzy systems with better than average performance.

S. Sushmita and S. Chaudhury [30] designed a case-based reasoning system for stock analysis in financial market, see also [31, 37], using hierarchical structure for case representation. The developed method also makes use of a multi-criteria decision-making algorithm which provides the most suitable solution with respect to the current market scenario. F. Cheong [31] describes a method for building HFS design with high input dimensions based the MacVicar-Whelan meta-rules. The method was tested on two applications: the Mexican and Argentinean currency exchange rates.

R.R. Yager [32] examines the basic assumptions in the Mamdani model. Fuzzy rules are considered as a partitioning of the input space. Different representations of the rule consequent are discussed. A new representation of fuzzy rules is introduced that is called the hierarchical prioritized structure (HPS). R. Holve [33] introduces a new method of rule generation for HFS called a hierarchical fuzzy associative memory (HIFAM). A HIFAM, structured as a binary tree, overcomes the curse of dimensionality when the number of inputs increases. R. Sindelar [34] describes a method for a fuzzy hierarchical structure design that uses data to design a structure of the fuzzy subsystems. The fuzzy structure is designed level by level from the data and therefore there is no need to develop an initial fuzzy model (single layer) avoiding HFS decomposition problem.

Y.J. Mon and C.M Lin [35] proposed a hierarchical fuzzy sliding-mode control to achieve asymptotic stability of the system. The nonlinear system is decomposed into several subsystems and the state response of each subsystem can be designed to be governed by a corresponding sliding surface. The

whole system is controlled by a hierarchical sliding-mode controller. The inverted pendulum system is used to test the proposed method. Later they improved their hierarchical fuzzy sliding-mode controller with decoupling of the nonlinear inverted pendulum system into several subsystems. Z.M. Yeh and K.H. Li [36] proposed a multistage control system for the inverted pendulum system that reduced the number of rules. R.J. Stonier and M. Mohammadian [37] presented introduction to hierarchical fuzzy control with the use of evolutionary algorithms on several examples: interest rate prediction, inverted pendulum, collision-avoidance in a robot system, micro-robot control, and co-evolutionary algorithm. D. Dasgupta [38] was one of the first researchers that explored evolving neuro-controllers for a dynamic system using genetic algorithms. W. Wang [39] designed a sliding mode controller for one-input multiple-output system where sliding surfaces are organized in a cascade thus creating a hierarchical system. F. Cheong and R. Lai [40] addressed problems with the use of hierarchical fuzzy logic controllers, especially in the automatic design of controllers. This includes the coordination of intermediary outputs (approximate controllers) of sub-controllers at lower levels of the hierarchy. The authors describe a method for the automatic design of a hierarchical fuzzy logic controller using an evolutionary algorithm called differential evolution. The method is developed for a wide class of control systems and the feasibility of the method is demonstrated by developing a two-layered HFS for controlling the inverted pendulum system. Other examples of hierarchical fuzzy control applied to the inverted pendulum system can be found in [41—47]. Related studies, but not applied to the inverted pendulum system, can be found in [47—58].

3 Inverted pendulum system

The inverted pendulum system consists of the cart and a rigid pole hinged to the top of the cart, see Fig. 1. The cart moves left or right on a straight bounded track and the pole swings in the vertical plane determined by the track. The dynamics of the system is modeled by the following equations:

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= u + m L (\sin(x_3) x_4^2 - \dot{x}_4 \cos(x_3)) / (M+m) \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= \frac{g \sin(x_3) + \cos(x_3) (u - m L x_4^2 \sin(x_3)) / (M+m)}{L (4/3 - m \cos(x_3)^2 / (M+m))}
 \end{aligned} \tag{1}$$

where x_1 is the position of the cart, x_2 is the velocity of the cart, x_3 is the angle of the pole, x_4 is the angular velocity of the pole, u is the control force on the cart, m is the mass of the pole, M is the mass of the cart, L is the length of the pole, and g is gravitational acceleration. The control force is applied to the cart to prevent the pole from falling while keeping the cart within the specified bounds on the track. The

system has the following parameters: $m = 0.1$ kg, $M = 1$ kg, $L = 0.5$ m, $g = 9.81$ ms⁻², with state variables restricted by: $-1.0 \leq x_1 \leq 1.0$ and $-\pi/6 \leq x_3 \leq \pi/6$.

Fuzzy controller task is to stabilise the system about the unstable reference position $\vec{x} = \vec{0}$ whilst maintaining the system within the target region (TR) defined by the following bounds: $|x_1| \leq 0.1$, $|x_2| \leq 0.1$, $|x_3| \leq \pi/24$, $|x_4| \leq 3.0$. The fuzzy controller is to ensure the state variables convergence to the TR and to maintain them within the TR for a prescribed time limit $T_f = 20.0$.

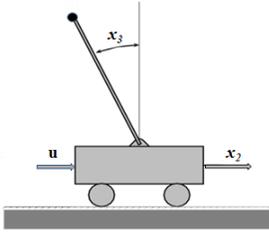


Fig. 1 Inverted pendulum system.

Each domain region for input variables x_i is divided into five overlapping intervals covered by membership sets A_i^k , $k = 1, \dots, 5$, encoded as integers from 1 to 5. The output variable u is divided into seven regions covered by membership sets B^k , $k = 1, \dots, 7$. All fuzzy membership functions are assumed to be triangular. Given a fuzzy rule base with M rules and n antecedent variables, a fuzzy controller as given by (1), with Mamdani product inference engine, uses a singleton fuzzifier and centre average defuzzifier to determine output variables:

$$u = \frac{\sum_{l=1}^M \bar{u}^l (\prod_{i=1}^n \mu_{A_i^l}(x_i))}{\sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i))} \quad (2)$$

where \bar{u}^l are centres of the output sets B^l and $\mu_{A_i^l}$ are membership functions associated with fuzzy sets A_i^l .

The following notation for 2-layered HFS topologies is used: $L2-mn-kl$ denotes that input variables for layer 1 are x_m, x_n and for layer 2: x_k, x_l , where all integers $m, n, k, l \in [1, 4]$. Similarly, for a three-layers HFS the following notation: $L3-mn-k-l$ denotes that input variables for layer 1 are x_m, x_n for layer 2: x_k , and for layer 3: x_l , where all integers $m, n, k, l \in [1, 4]$. For other 2 and 3-layers HFS the same notation is used, for example: $L3-3-4l-2$ means that input in layer 1 is x_3 , in layer 2: x_4, x_l , and in layer 3: x_2 .

Generally speaking, the number after capital letter L defines the number of layers, followed by groupings of numbers separated by hyphens representing input variables in every layer.

4 Hierarchical fuzzy system

The structure of the hierarchical fuzzy system for the inverted pendulum problem defines the fuzzy controller to be found (by the EA). This means finding the controller input configuration and fuzzy rule bases (IF-THEN rules) corresponding to each component of the HFS. In the HFS structure the single rule base is split into separate rule bases corresponding to each layer of the hierarchical structure, see for example Fig. 2 – 7. The hierarchical fuzzy structure can be encoded as a string of integer or real numbers. If binary approach is used, encoding the HFS by string of 0-1 numbers, then well developed theory of genetic algorithms can be used to find a control problem solution. In this paper we use HFS encoding by integer numbers, and the search technique is called evolutionary algorithm.

In the context of HFS, topology means both structure (layers) and input configuration of the hierarchical fuzzy system. In inverted pendulum problem there are four possible structures: 1-layer, 2-layers, 3-layers, and 4-layers, with different input configuration (except single layer topology). The number of rules in the hierarchical fuzzy system is a linear function of the number of input variables. Let assume that there are n input variables in a L -layers structure. For every input variable there are m fuzzy sets associated with that variable. Assume further that in the first layer there is n_1 input variables, $2 \leq n_1 < n$, and $n_i + 1$ in the i -th layer, $n_i \geq 1$. If $n_1 = n_i + 1 = c$ is constant for $i = 2, \dots, L$, then the total number of rules in the hierarchical fuzzy system is given by: $M = \frac{m^c}{c-1} (n - 1)$. If $m \geq 2$ the number of rules M is minimized when $c = 2$, which means that there are two input variables in every layer [51]. This input configuration provides the minimal number of fuzzy rules in the knowledge base but it does not necessarily provide the best configuration from the control system perspective. Often, such systems do not provide sufficient control performance, especially in complex high-dimensional systems. The decomposition needs to be performed along the weak interdependencies between input variables. Obviously, it requires certain knowledge of the physical system in the absence of any automated method of HFS decomposition. Note, that only $L3-mn-k-l$ configuration satisfies the above condition. Topologies $L2-mn-kl$ do not provide the minimal size for the knowledge base.

In general, the first step in establishing the hierarchical structure for a given dynamical system is to establish the importance of its input variables and their interactions. If the character of the dynamical system allows it, the variables should be grouped according to the degree they influence the output of the system and their inter-relations. This allows designing the layered structure and input configuration. The structure can be a combination of strictly hierarchical layers and parallel layers with similar input variables grouped according to their inter-relations, see example of such a topology in Fig. 2.

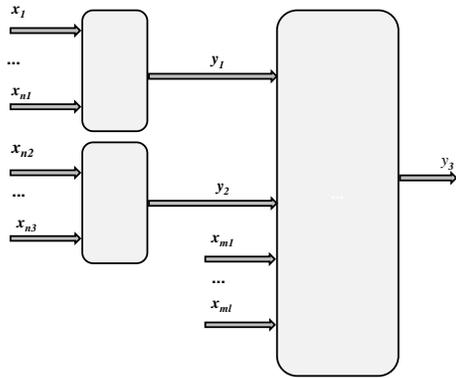


Fig.2 Example of HFS with two parallel layers.

4.1 Two layered HFS

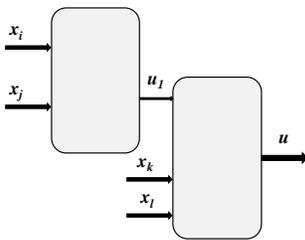


Fig.3 HFS: 2-layers input configuration.

There are six different topologies of the 2-layered hierarchical fuzzy system in the $L2-mn-kl$ configuration: two input variables in the first layer and two input variables plus intermediary control variable from layer 1 in layer 2. This decomposition does not exhaust all possibilities as different input configurations can be considered, for example: three input variables in layer 1 and one input variable plus intermediary control from layer 1 in layer 2. Another possibility is to have one input in layer 1 and three input variables plus intermediary control in layer 2 which would result in knowledge base consisting of 880 rules which is larger in size than single layer architecture. The architecture of 2-layered HFS is shown in Fig. 3.

For the inverted pendulum system the first knowledge base has the two inputs to produce as output a first approximation of the control u_1 . This u_1 together with another state input x_i and x_j , $i, j \in [1,4]$ are used as input in the second knowledge base to produce the final control output u . In the first layer there are $25 = 5^2$ rules in the knowledge base. The l^{th} fuzzy rule for the first layer has the form: *If* (x_i is A_i^l) *and*

$(x_j \text{ is } A_j^l)$ Then $(u_l \text{ is } B^l)$, where A_k^l , $k = 1,2,3,4$, are fuzzy sets for input variables x_k , $k = 1,2,3,4$, respectively, and where B^l are fuzzy sets for output variable u_l . For the second layer there are $175 = 7(5)^2$ rules in the knowledge base. The l^{th} fuzzy rule for the second layer has the form: *If* $(u_l \text{ is } C^l)$ *and* $(x_i \text{ is } A_i^l)$ *and* $(x_j \text{ is } A_j^l)$ *) Then* $(u \text{ is } B^l)$, where C^l are fuzzy sets for the input control variable u . There are a total of 200 fuzzy rules in this hierarchical structure while there are 625 rules in the single layer rule base: $5^4 = 625$.

4.2 Three layered HFS

In $L3-mn-k-l$ configuration there are twelve different topologies for the 3-layered hierarchical fuzzy system with two input variables in the first layer, and one input variable plus intermediary control in layer 2 and 3. Again, this decomposition does not exhaust all possibilities, as different input configurations can be considered but with an increased number of rules in the knowledge base. The architecture of 3-layered HFS is shown in Fig. 4. This input configuration provides the minimal number of fuzzy rules in the knowledge base.

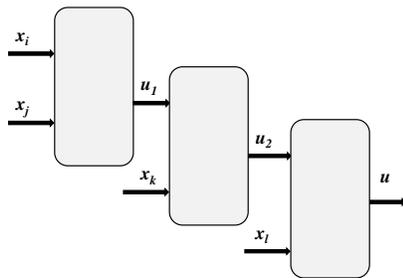


Fig. 4 HFS: 3-layers input configuration.

For this system the first knowledge base has the two inputs, x_i and x_j , $i, j \in [1,4]$ to produce as output a first approximation of the control u_1 . This u_1 together with x_k are used as input in the second knowledge base. Then the second layer produces another approximation of control u_2 which with x_l is used as input to the third (and final) layer to produce the final control output u . There are a total of 95 fuzzy rules in this hierarchical structure.

4.3 Four Layered HFS

There are twenty four different topologies for the 4-layered HFS for the inverted pendulum with one input in every layer plus intermediary control in layer 2, 3, and 4. The architecture of 4-layered HFS is shown in Fig. 5.

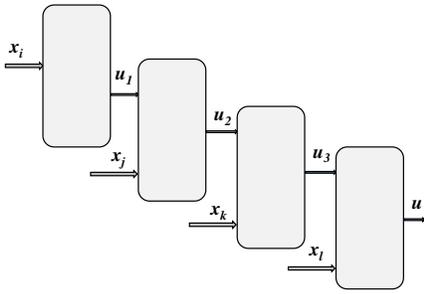


Fig. 5 HFS: 4-layers input configuration.

In the first layer there are only five rules in the knowledge base. The l^{th} fuzzy rule for the first layer has the form: *If* (x_i is A_i^l) *Then* (u_1 is B^l), where $A_k^l, k = 1,2,3,4$ are fuzzy sets for input variables $x_k, k = 1,2,3,4$, respectively, and where B^l are fuzzy sets for output variable u_1 . For all the other layers there are $35 = 7 \cdot 5$ rules in their respective knowledge bases. For the second layer the l^{th} fuzzy rule has the form: *If* (u_1 is C^l) *and* (x_k is A_k^l) *Then* (u_2 is B^l), where C^l are fuzzy sets for the input control variable u_1 . Fuzzy rules for the third and fourth layer has a similar form. There are a total of 110 fuzzy rules in the 4-layered hierarchical structure.

5 Alternative topologies

There are other input configurations within two and three layered hierarchical structures that do not have two inputs in the first layer (one of the conditions for the minimal size of the rule base). Two examples of 2-layered alternative input configurations are shown in Fig. 5. In the first configuration there are three input variables in layer 1 and one input variable plus intermediary control in layer 2 which results in 160 rules in the knowledge base. In the second configuration there is one input in layer 1 and three input variables plus intermediary control in layer 2 which results in a knowledge base of 880 rules - more than in a single layer knowledge base. Two 3-layered alternative topologies are shown in Fig. 6. Both configurations have 215 rules in their rule bases.

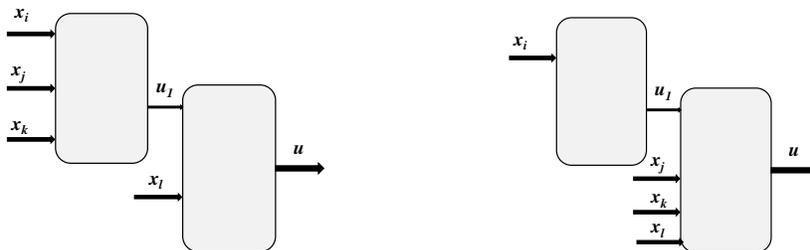


Fig. 6 HFS: 2-layers alternative input configurations.

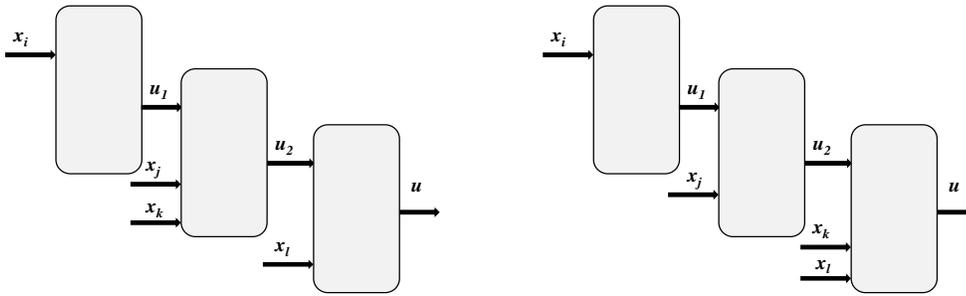


Fig. 7 HFS: 3-layers alternative input configurations.

6 Proposed evolutionary algorithm based approach

The evolutionary algorithm is a heuristic search technique that maintains a population of individuals. Each individual can be considered to represent a potential solution to a given problem. Each individual is assigned a measure of fitness which determines how accurate it is as a potential solution to the problem. The new population is obtained from the old one by the use of genetic operators such as crossover and mutation. An elitism strategy is used to pass the fittest individuals to the new population, so that the information encapsulated in the best individual is not lost and passed to the next generation. A selection process is used to obtain parents for mating in the current generation. The most popular is proportional selection to select randomly two parents based on their fitness in proportion to the overall total fitness of the population. Another is tournament selection in which a specified number of possible parents are selected at random from the population. A tournament is then held to select the two fittest strings and they are used as parents in the next process of crossover to generate children to be passed into the next generation.

In the crossover operation a number of 'parent' strings, typically two, are recombined to create 'child' strings. The most popular crossover operator is the one-point, arithmetic, and uniform crossover. The crossover operator plays a role of sexual reproduction in which two individuals exchange parts of their strings to produce offspring.

With a given probability the mutation operator mutates elements of the individual in the population. This ensures satisfactory diversity within the population which is required for the EA to find better approximate solutions to the problem.

With an appropriate selection of EA parameters and operators, the algorithm is allowed to evolve. It is terminated when pre-defined termination condition is satisfied; usually at a fixed number of generations

or until there is minimal change or no change to the string which has the best fitness. The fittest individual is taken as the best possible solution learnt by the algorithm.

The HFS decomposition is not unique and may give rise to variables with no physical significance. This can cause difficulties in obtaining a complete class of rules from human experts. Therefore, the rules need to be learnt by some learning algorithm, for example an evolutionary algorithm. These fuzzy rules are typically evolved with no previous knowledge other than input-output data, or the physical system model.

In the knowledge base of any layer each fuzzy rule is uniquely defined by the position of the consequent part in the string. The consequent part is identified by a particular output fuzzy set, for example, B^k . Such a fuzzy set can be identified by the integer $k \in [1,7]$. For example, 2-layered fuzzy rule base can be represented as a linear individual string of $M = 25 + 175 = 200$ consequents, $\vec{p}_k = (a_1, \dots, a_{200})$, where a_j is an integer $\in [1,7]$ for $j = 1, \dots, 200$. Other hierarchical fuzzy system structures can be represented in a similar fashion.

Full replacement policy (constant population size) is used and tournament selection with size $n_T = 4$. A strong elitism policy is implemented: four copies of the top five individuals are passed to the next generation. To maintain diversity of the population crossover operators of parent strings to form two children in the next generation are used. In examination of different topologies so called random crossover is implemented. The random crossover procedure creates *child1* from *parent2* by copying it, then randomly selecting m -genes in the parent1 string to copy them in the corresponding positions in the *child1* string. The procedure is repeated for the *child2* string with parent strings roles reversed. The random crossover operator gives more control over crossover process as the number of genes subject to exchange can be arbitrarily determined. Mutation is undertaken with probability p_m whose value is determined by a mutation schedule that decreases from 0.8 to 0.001 over 300 generations.

```

if ( gen ≥ 0 & gen < 50 ) pm = 0.8
if ( gen ≥ 50 & gen < 100 ) pm = 0.7
if ( gen ≥ 100 & gen < 150 ) pm = 0.6
if ( gen ≥ 150 & gen < 200 ) pm = 0.3
if ( gen ≥ 200 & gen < 250 ) pm = 0.1
if ( gen ≥ 250 & gen < 300 ) pm = 0.01
if ( gen > 300 ) pm = 0.001

```

where *gen* denotes the generation number. The operator is defined by the following pseudo code:

```

if (mutate) {
  if (ak = 7) ak = ak - rnd(1,3)
  else if (ak = 1) ak = ak + rnd(1,3)
  else if (flip(0.5)) ak = ak + rnd(1,3)
    else ak = ak - rnd(1,3)
  if (ak > 7) ak = 7

```

if (ak < 1) ak = 1 }

The objective function is evaluated as follows: each string \vec{p}_k is decoded into two or more components defining the fuzzy knowledge base for each layer, then the formula (2) is used to evaluate u_1 , u_2 , and u_3 to find the final control to be applied at each value of the state \vec{x} . The system state equations are integrated by the Runge-Kutta algorithm with step size 0.02. The fitness f_k to be minimised, is then calculated based on certain measures of the behaviour of the system: the accumulated sum of normalised absolute deviations of x_1 and x_3 from zero, the average deviation from vertical, the average deviation from the origin or $T - T_S$ where T_S , the survival time, is taken to mean the total time before the pole and cart break defined bounds. A penalty is added to the objective if the final state breaks the following bounds: $|x_1| \leq 0.1$, $|x_2| \leq 0.1$, $|x_3| \leq \pi/24$, $|x_4| \leq 3.0$, i.e., leaves the designated target region. The objective function is defined as: $f_k = \omega_1 F_1 + \omega_2 F_2 + \omega_3 F_3 + \omega_4 F_4 + \omega_5 F_5$

$$\text{with: } F_1 = \frac{1}{N} \sum_1^N \frac{|x_1|}{x_{max}}, F_2 = \frac{1}{N} \sum_1^N \frac{|x_2|}{\dot{x}_{max}}, F_3 = \frac{1}{N} \sum_1^N \frac{|x_3|}{\theta_{max}}, F_4 = \frac{1}{N} \sum_1^N \frac{|x_4|}{\dot{\theta}_{max}}, F_5 = \frac{1}{N} (T - T_S),$$

where $x_{max} = 1.0$, $\theta_{max} = \pi/6$, $\dot{x}_{max} = 1.0$, $\dot{\theta}_{max} = 3.0$. N is the number of iteration steps. Survival time is defined as: $T_S = 0.02 \cdot N$, with $T = 0.02 \cdot N_{max}$, where the max. number of iterations $N_{max} = 1000$.

7 Experiments and analysis of results

The relatively low number of possible topologies for the inverted pendulum enables their examination one by one and finding the topology with the best controller performance. The best controller is decided arbitrarily by considering its performance: time to reach target region, magnitude of control, and oscillations. For each topology ten simulations are run with randomly generated initial populations. The initial population $P(0) = \{ \vec{p}_k : k = 1, \dots, M_p \}$ is determined by choosing the a_j as a random integer in $[1,7]$. Population size is set at $M_p = 500$. The initial state is: $\vec{x}_0 = (0.5, 0.0, 0.01, 0.0)$. The EA is terminated after 300 generations. The following fitness function parameters: $\omega_1 = 3000$, $\omega_2 = 2000$, $\omega_3 = 0$, $\omega_4 = 0$, $\omega_5 = 5000$, are selected for all simulations except single layer fuzzy system with: $\omega_1 = 1000$, $\omega_2 = 0$, $\omega_3 = 1000$, $\omega_4 = 0$, $\omega_5 = 5000$. Selecting different weights ω introduces a bias in the EA towards one or a group of input variables. In most simulation, the most difficult was to control the cart's position and therefore ω_1 was assigned a larger value than other weights, except ω_5 that corresponds to the survival time. Controller acts on the system for $T = 20$.

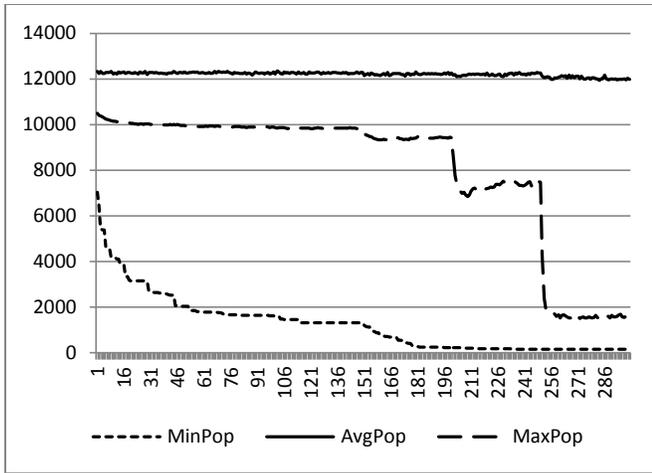


Fig. 8 Minimum, average and maximum objective function values over 300 generations for *L2-34-12*.

The minimum, average and maximum of objective function are examined for every topology over consecutive generations. The results are fairly similar, both for the 2 and 3-layered HFS. Examples for typical simulations are given in Fig. 8.

Comparison between controllers with *L2-14-23* topology and *L2-23-14* shows that swapping input variables between layer 1 and layer 2 can have a significant effect on the controller performance. This indicates that the HFS topology is a decisive factor in the controller performance. This assertion is confirmed by the 3-layered HFS results presented in the following section.

7.1 One Layer Topology Results

The EA is run ten times for the single layer FS with different initial random populations. Within around 200 generations the best fuzzy controller at each generation achieved convergence of state variables to the designated TR and maintained it within this region for the remainder of the prescribed time T_f . Typical convergence and controller output values are shown in Fig. 9. As it can be seen from Fig. 9 the stabilisation is smooth and regular for all state variables. The controller is ‘frugal’, with values lying in $[-0.83, 0.202]$, the best simulation results in terms of control magnitude. In ten simulations for *L1-1234* there are controllers with faster stabilisation times but with higher control magnitude.

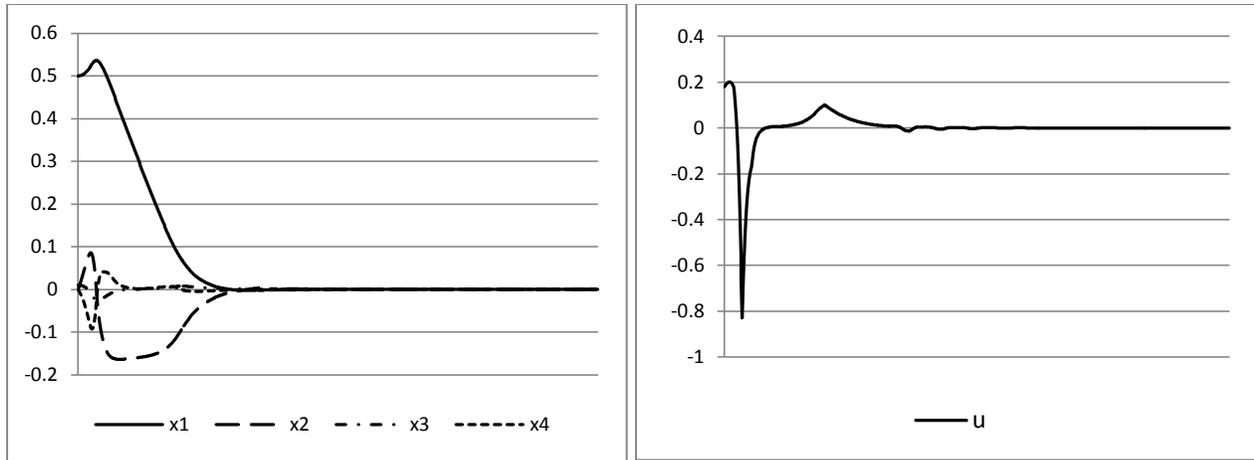


Fig. 9 State variables convergence (left) and controller (right) *L1-1234*.

7.2 Two Layers Topology Results

The EA converged on average to a satisfactory solution within 100—184 generations. The best result is shown in Fig. 10, where pole angle x_3 and its angular velocity x_4 are input to layer 1, and cart position x_1 and its speed x_2 as input variables to layer 2. This result demonstrates that the inverted pendulum system should be decomposed into two input variables groupings: cart variables: x_1 and x_2 (cart's position and its velocity) and pole variables: x_3 and x_4 (pole angle and pole's angular velocity). Stabilisation is very quick and to 5 decimal place accuracy for all variables. This result gives the first indication as to which input configuration provides the best controller performance. Furthermore a small control effort is required with control values in $[-2.0, 1.6]$.

Obviously, if topology *L2-34-12* provides good control structure then *L2-12-34* controller is expected to achieve similar performance as it is decomposed along the same weak interdependence between input variables only with pole variables replaced with cart variables as input in the first layer. Indeed, the *L2-12-34* controller performance is similar in both state variable convergence and controller magnitude to *L2-34-12* controller. However, results for controllers with topology *L2-12-34* were much less consistent in ten simulation runs than for *L2-34-12*. On average, controller with topology *L2-34-12* provides slightly better performance and consistency of the EA solutions.

Table 1 Stabilisation times for 1 and 2-layered HFS

Run No	L1-1234	L2-12-34	L2-13-24	L2-14-23	L2-23-14	L2-24-13	L2-34-12
10	3.92	9.12	4.16	3.1	9.02	9.36	2.74

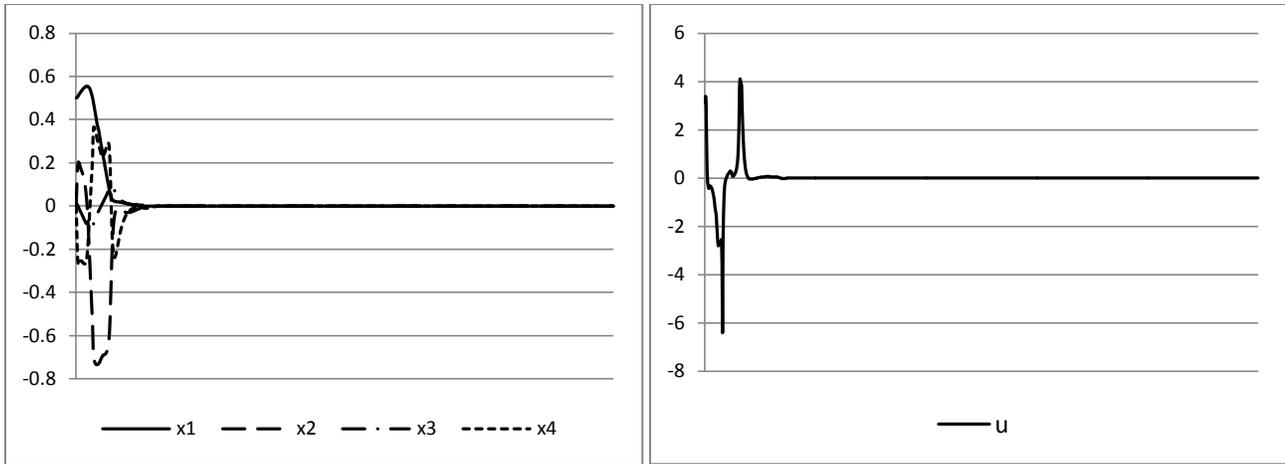


Fig. 10 State variables convergence and controller $L2-34-12$.

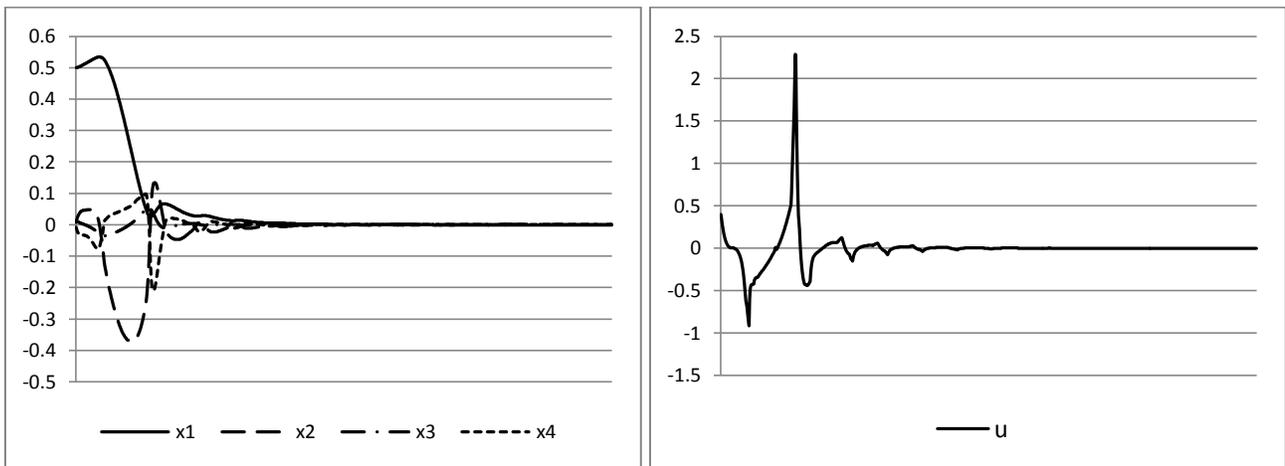


Fig. 11 State variables convergence and controller $L2-14-23$.

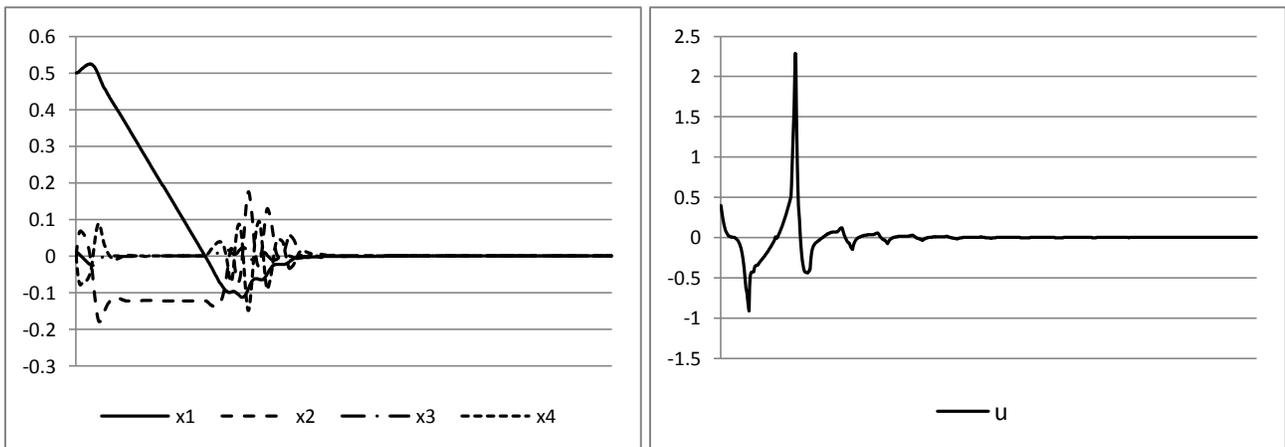


Fig. 12 State variables convergence and controller $L2-23-14$.

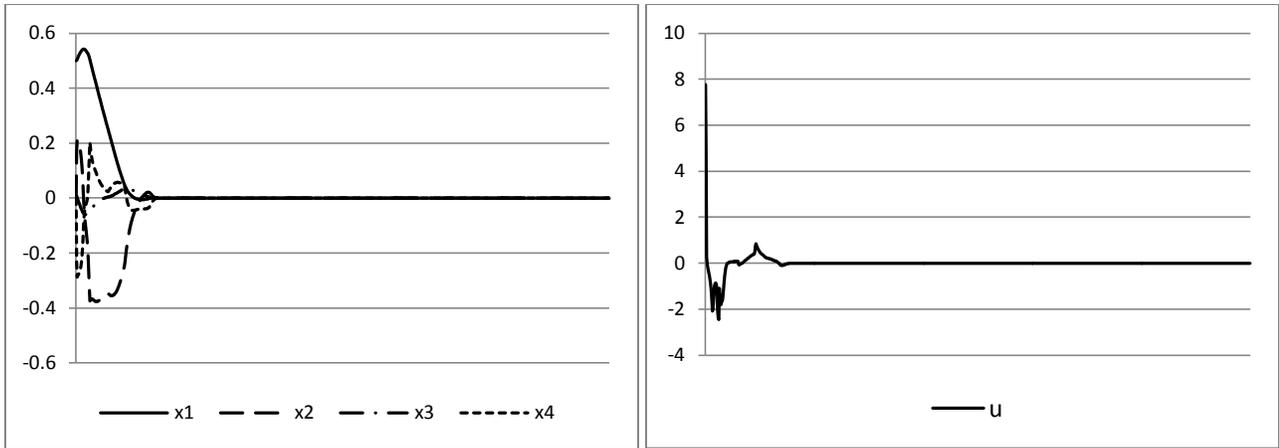


Fig. 13 State variables convergence and controller *L3-34-2-1*.

Comparison between controllers with *L2-14-23* topology and *L2-23-14*, see Fig. 11 and 12, shows that swapping input variables between layer 1 and layer 2 has a significant effect on the controller performance, which indicates that the HFS topology is a decisive factor in the controller performance. This assertion is confirmed by the 3-layered HFS results. The relatively good performance delivered by the controller with topology *L2-14-23* is an example of a ‘mixed’ input configuration where decomposition breaks strong interdependence of state variables. A poor controller performance can be expected in such cases but if most significant state variable is an input in the first layer then it has positive moderating effect on the control process.

Table 2 Learning Speed: 1 and 2 layered Topologies

Run No	<i>L1-1234</i>	<i>L2-12-34</i>	<i>L2-13-24</i>	<i>L2-14-23</i>	<i>L2-23-14</i>	<i>L2-24-13</i>	<i>L2-34-12</i>
1	201	150	119	183	150	201	13
2	181	93	174	204	170	188	160
3	204	139	157	179	167	160	114
4	180	54	111	222	150	152	103
5	202	124	151	161	138	176	154
6	217	126	139	199	7	106	169
7	205	127	211	158	150	153	52
8	201	114	159	188	155	158	150
9	189	30	153	167	112	205	177
10	199	44	169	203	151	227	172
Average	198	100	154	186	135	173	126

Table 3 Learning speed: 3-layered HFS

Run No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
1	28	153	210	166	109	192	144	202	154	204	81	55
2	157	93	181	193	150	188	212	165	3	160	178	57
3	156	103	172	115	4	171		179	153	156	161	157
4	151	119	202	211	154	205	162	163	162	162	143	210
5	164	150	201	176	112	236		167	164	244	68	94
6	133	164	205	185	152	191	209	165	150	200	146	81
7	150	1	206	158	117	169	159	208	23	195	152	27
8	151	151	151	49	10	137		36	145	177	151	204
9	152	62	233	174	145	170		184	157	166	156	150
10	152	142	213	153	152	210	215	123	130	155	207	129
Average	139	114	197	158	111	187	184	159	124	182	144	116
Total Avg	151											

where C1 denotes: $L3-12-3-4$, C2: $L3-12-4-3$, C3: $L3-13-2-4$, C4: $L3-13-4-2$, C5: $L3-14-2-3$, C6: $L3-14-3-2$, C7: $L3-23-1-4$, C8: $L3-23-4-1$, C9: $L3-24-1-3$, C10: $L3-24-3-1$, C11: $L3-34-1-2$, C12: $L3-34-2-1$.

7.3 Three Layers Topology Results

For the 3-layers HFS, the evolutionary algorithm converged to a satisfactory solution within 150—200 generations. Controller defined by topology $L3-34-1-2$ achieves all state variables smooth convergence to the origin without any large oscillations. The magnitude of control is reasonably low, within range of $[-2.4, 1.4]$. It is one of the best controllers among the 3-layers HFS. However, in ten simulations the controllers $L3-34-2-1$ generally outperformed the controllers $L3-34-1-2$ and therefore should be considered as the best controller amongst the 3-layers HFS, see Fig. 13. Note, that the state variables convergence to the TR is faster for $L2-34-12$ than for $L3-34-1-2$.

As can be seen from comparison of $L3-13-2-4$ and $L3-13-4-2$ controllers, a seemingly insignificant change in input configuration in layer 2 and layer 3 results in a significant change in the controller performance, see Fig. 14 and 15. A similar effect can be seen by in topologies $L3-14-2-3$ and $L3-14-3-2$. Three-layered topology breaks strong interdependence between state variables in layers 2 and 3. This does not have adverse effect on the controller performance for the ‘best’ topologies $L3-34-1-2$ and $L3-34-2-1$ because decomposition reflects physical properties of the system. However, for $L3-12-3-4$ or $L3-12-4-3$ it has slightly detrimental effect because the HFS decomposition breaks state variables interdependence. In some other cases, for example $L3-14-2-3$ or $L3-14-3-2$ it has a profound effect. Physical properties of the system under consideration require grouping of the input variables along weak state variables interdependence. In case of the inverted pendulum this grouping corresponds to two

subsystems: the cart represented by x_1 and x_2 , and the pole represented by x_3 and x_4 . Swapping the input variables between the layers but preserving to some extent abovementioned groupings has little effect on the controller performance. When this grouping principle is broken, the results are often detrimental (depending which variables are more influential in the dynamical system). In case of $L3-14-2-3$ and $L3-14-3-2$ it seems that controlling the angle of the pole is more crucial than controlling the cart's velocity as it is reflected in both topologies.

Table 4 Stabilisation times for 3-layered HFS

Run No	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
10	9.18	13.72	2.52	2.54	3.68	3.5		2.14	12.2	6.96	2.9	2.06

where C1 denotes: $L3-12-3-4$, C2: $L3-12-4-3$, C3: $L3-13-2-4$, C4: $L3-13-4-2$, C5: $L3-14-2-3$, C6: $L3-14-3-2$, C7: $L3-23-1-4$, C8: $L3-23-4-1$, C9: $L3-24-1-3$, C10: $L3-24-3-1$, C11: $L3-34-1-2$, C12: $L3-34-2-1$

7.4 Four Layers Topology Results

Four variants of the 4-layered topology: $L4-3-4-1-2$, $L4-3-4-2-1$, $L4-4-3-1-2$, and $L4-4-3-2-1$ are examined. The last two topologies produced good controller performance. One of the controllers representing topology $L4-3-4-1-2$ produced a low control magnitude in range $[-2.0, 2.8]$, see Fig. 17.

Examination of 4-layered topologies can determine which input variables are most influential in the inverted pendulum system. It was found that topologies $L2-34-12$ and $L3-34-2-1$ are the best performing controllers. The simulation results show that the topology $L4-4-3-2-1$ is the most consistent in producing well performing controllers for ten different initial populations with $L4-4-3-1-2$ close behind. This clearly indicates that the most influential input variable is the angular speed of the pole x_4 , second - the angle of the pole x_3 , and then cart's speed x_2 and its position x_1 .

It is noted that 4-layered topologies are significantly outperformed by lower layers HFS controllers. This indicates that for the inverted pendulum problem this ladder-like structure does not produce well performing controllers.

Table 5 Stabilisation times for 4-layered HFS

$L4-3-4-1-2$	$L4-3-4-2-1$	$L4-4-3-1-2$	$L4-4-3-2-1$
3.22	1.68	2.64	1.82

7.5 Alternative Topologies Results

Simulations conducted for 2 and 3-layered HFS that do not have two inputs in the first layer are shown in Fig. 16—20. Variations on topologies $L2-34-12$ and $L3-34-1-2$ were selected for simulations as they were shown to be the best controller candidates. The following input configurations were considered:

L2-3-412, *L2-341-2*, *L3-3-41-2* and *L3-3-4-12*. The controller performance is better than expected, especially for topologies *L3-3-41-2* and *L3-3-4-12*, with smooth control and relatively low control magnitude.

Considering the performance of these controllers they cannot be ruled out on the basis of their large rule bases. The large number of rules does not hamper performance even though it might cause problems in controller application if controller speed is of critical importance.

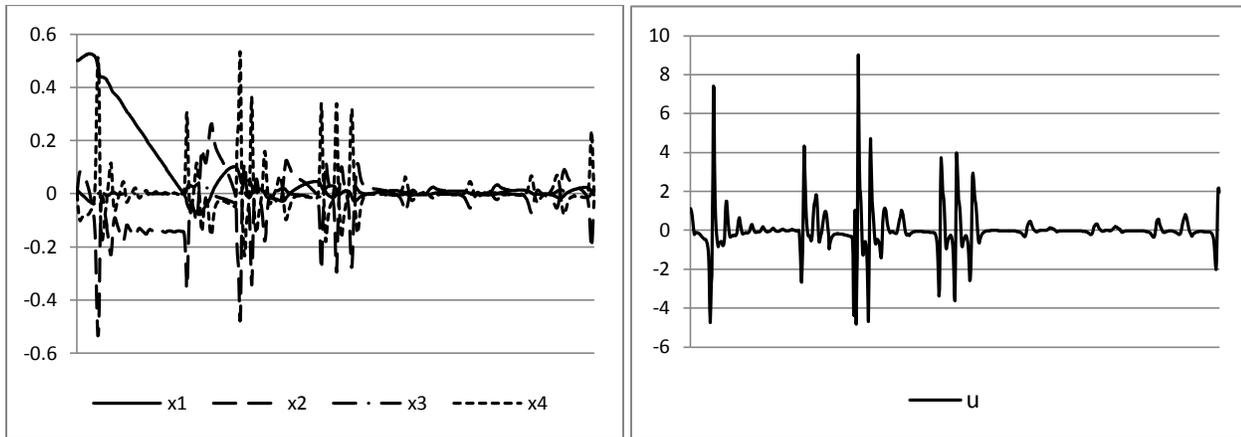


Fig. 14 State variables convergence and controller *L3-13-2-4*.

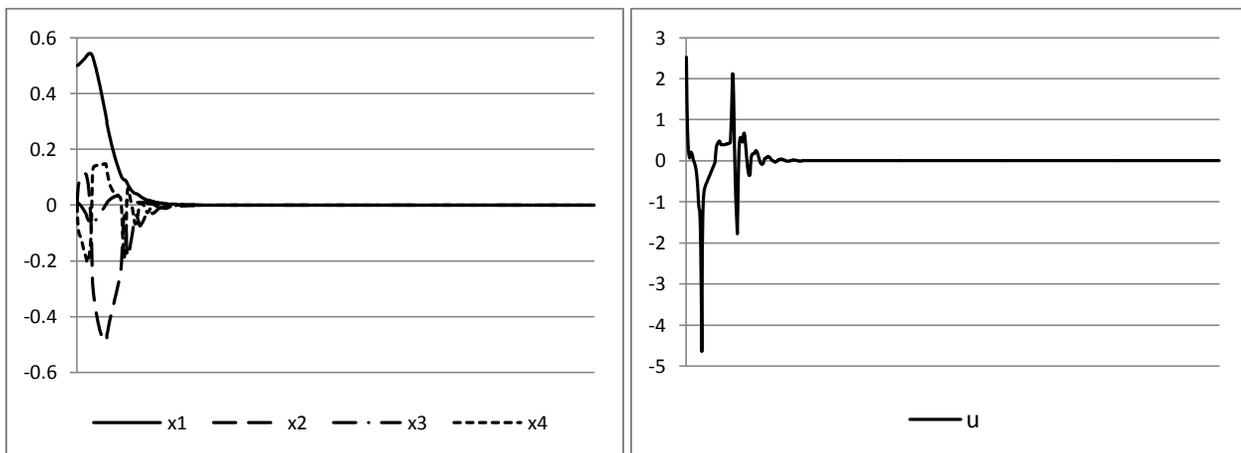


Fig. 15 State variables convergence and controller *L3-13-4-2*.

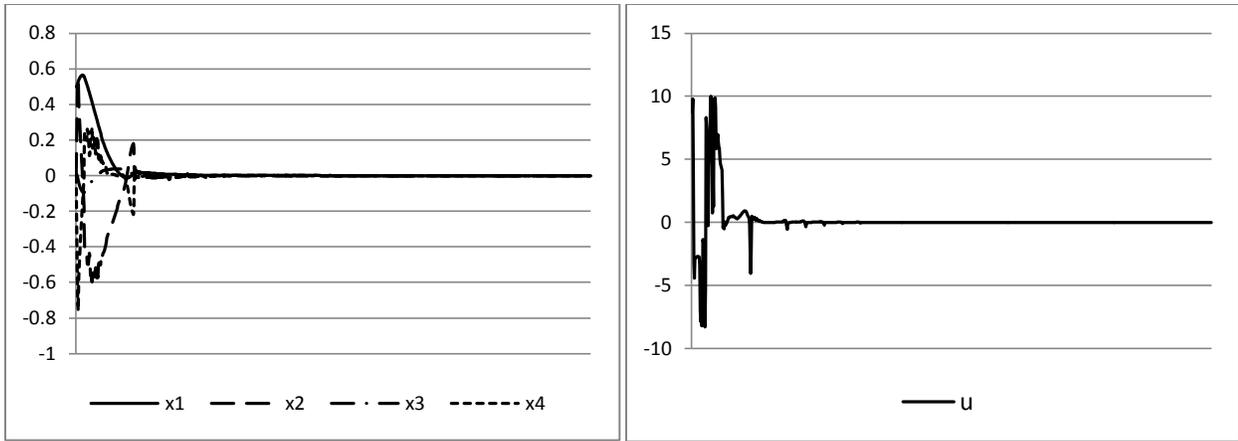


Fig. 16 State variables convergence & controller *L4-4-3-2-1*.

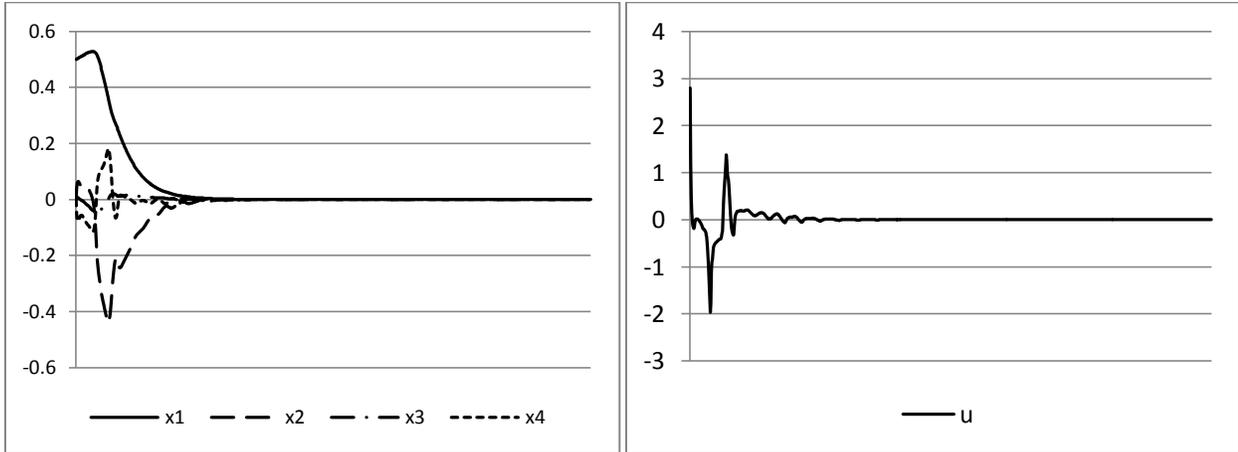


Fig. 17 State variables convergence & controller *L4-4-3-1-2*.

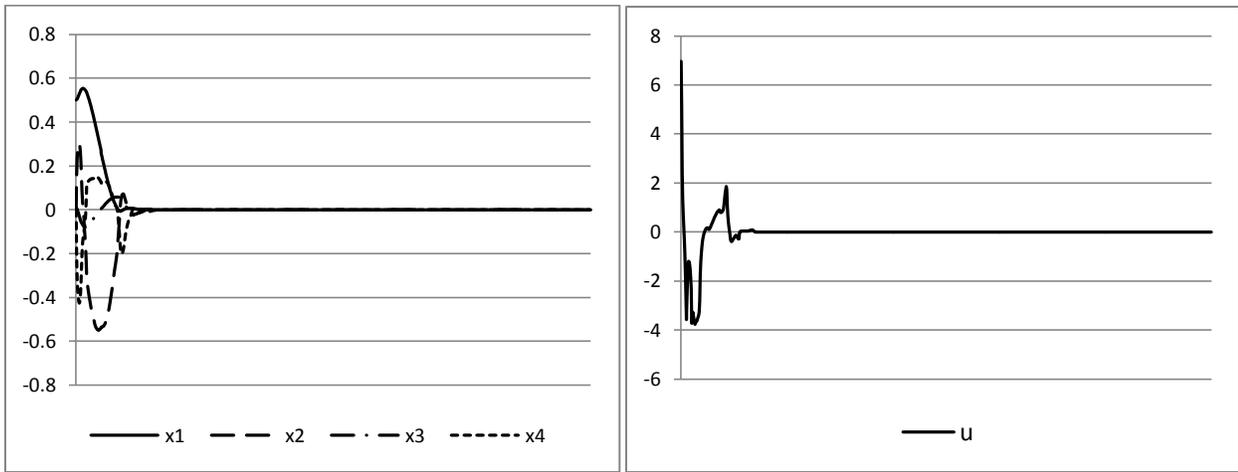


Fig. 18 State variables convergence & controller *L4-3-4-2-1*.

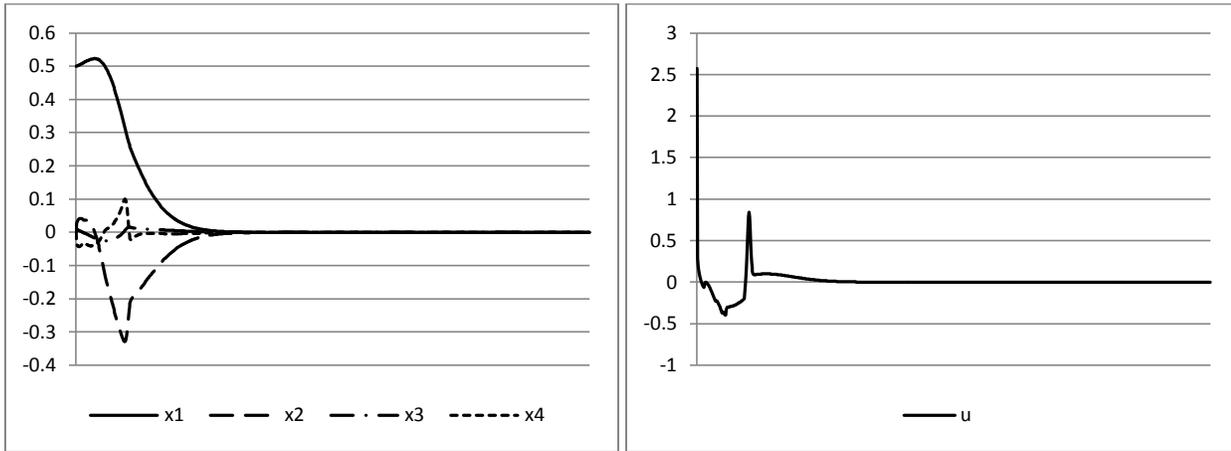


Fig. 19 State variables convergence & controller $L4-3-4-1-2$.

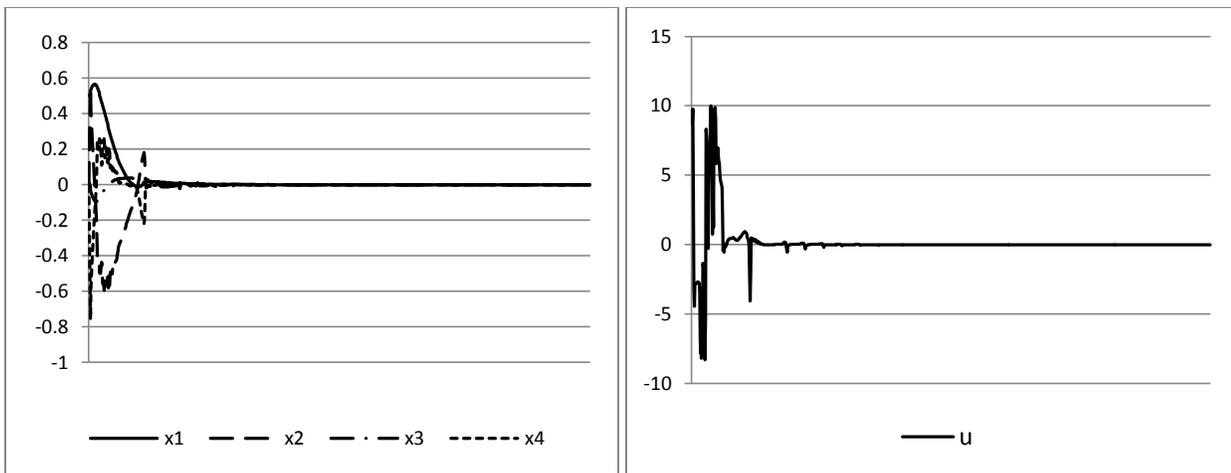


Fig. 20 State variables convergence & controller $L4-4-3-2-1$.

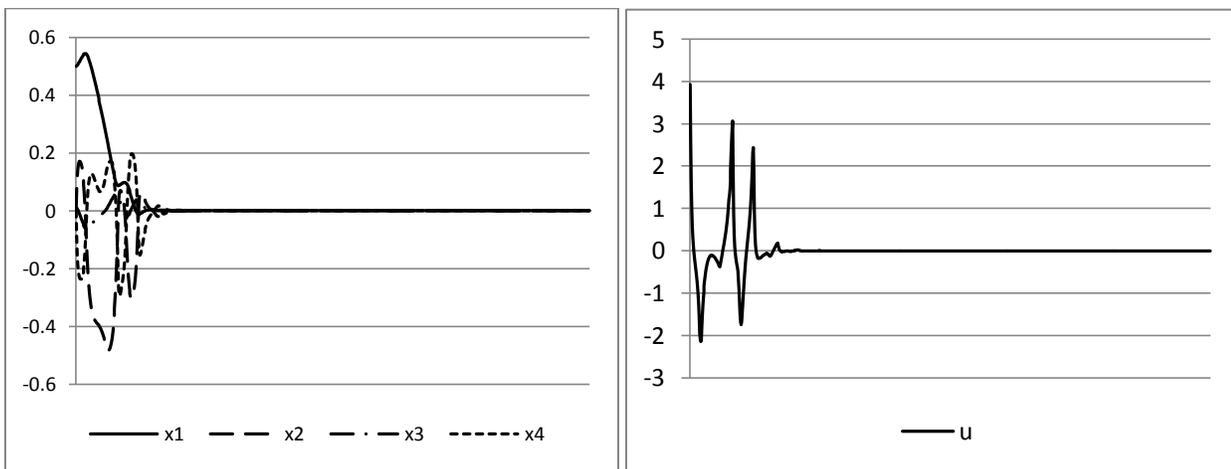


Fig. 21 State variables convergence and controller $L2-3-412$.

8 Controller test

Intermediate variables u_1 and u_2 may have not any physical representation. They can be considered as approximations to the controller action. By testing their performance, with one or two layers removed from the HFS one can expect to determine the robustness of the controller. Examining the approximate control provides insight into ‘working parts’ of the HFS. When one layer is eliminated it is denoted by $L2-mn$ or $L3-mn-k$. In a 3-layered HFS $L3-mn$ denotes elimination of the two last layers. For controller tests the best controllers for the 2 and 3-layered HFS are selected. The best 2-layered HFS are: $L2-34-12$, $L2-14-23$, $L2-13-24$ and $L2-12-34$. The best 3-layered HFS are: $L3-34-1-2$, $L3-14-3-2$, and $L3-13-4-2$.

8.1 Two layers HFS controller test results

The best performing among the 2-layered HFS is the controller with topology $L2-34-12$. The controller is tested with its last layer removed: topology $L2-34$. The controller stabilises the system for 1.6 time units before ‘crashing’, i.e., until breaking the state bounds: $|x_1| \leq 1.0$ and $|x_3| \leq \pi/6$. Otherwise the controller exhibits very regular behavior, see Fig. 22. Then the controller with topology $L2-14-23$ is tested with its last layer removed. One layer version $L2-24$ ‘crashed’ at $t = 0.8$, the angular velocity of the pole x_4 rising steeply, see Fig. 23.

The controller with topology $L2-13-24$ is tested with its last layer removed. As can be seen in Fig. 24 the approximate controller with only one layer rule base attempts to stabilise the system for the whole period of time $T_f = 20$: the angle of the pole x_3 is stabilised in a narrow band around the origin, the angular velocity x_4 oscillates but the values of x_4 remain within $[-1, 1]$ band. Therefore, the approximate controller, with second layer removed, performs well, while the controller with full rule base performs poorly.

The controller with topology $L2-12-34$ with its last layer removed is one of the worst performing controllers, see Fig. 25. This poor performance comes in spite of relatively good performance of the controller with the full rule base. The controller stabilises the system for 3.76 time units until the angle of the pole x_3 breaks the state limits.

8.2 Three layers HFS controller test results

In the 3-layered HFS the controllers are tested by removing the last layer or the two last layers. The controller with topology $L3-34-1-2$ is analysed first. The simulation results are shown in Fig. 26—27. The approximate controller u_2 maintains control of the system for the whole prescribed time $T_f = 20$ and

exhibits very regular behaviour. The controller with topology $L3-23-4-1$ is analysed and simulation results are shown in Fig. 28—29. The control pattern is regular but the ‘crash-time’ for this controller is relatively short (1.2 and 0.78 respectively).

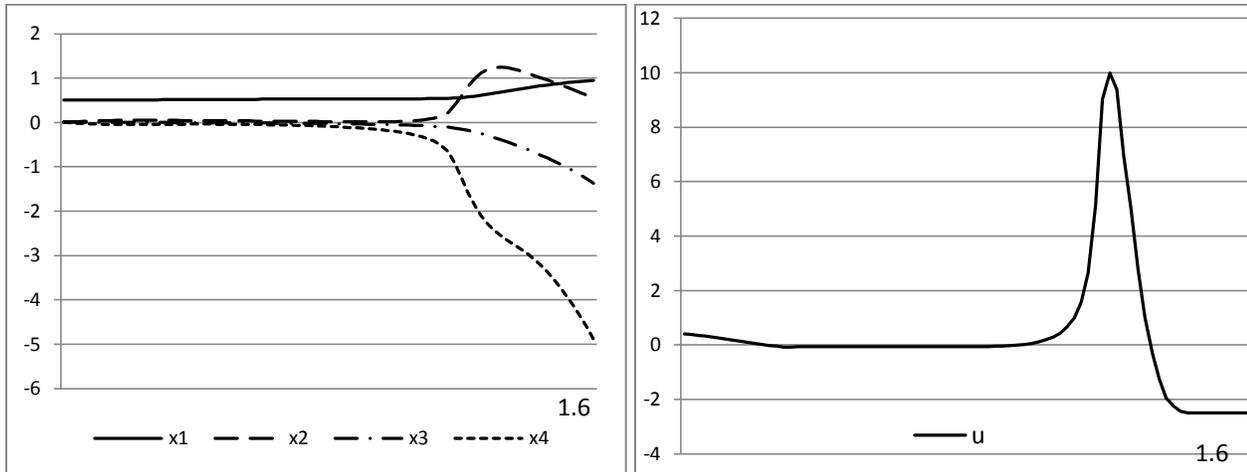


Fig. 22 State variables convergence and controller $L2-34$.

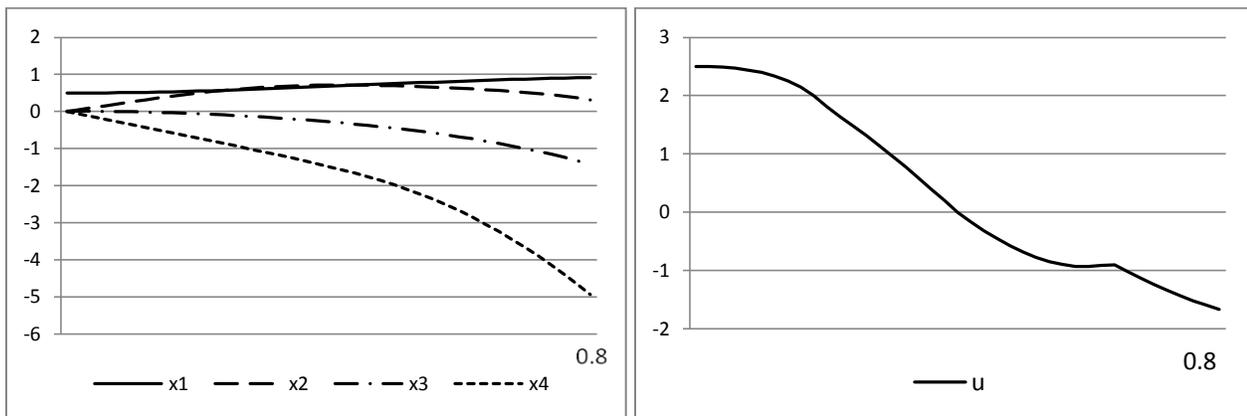


Fig. 23 State variables convergence and controller $L2-14$.

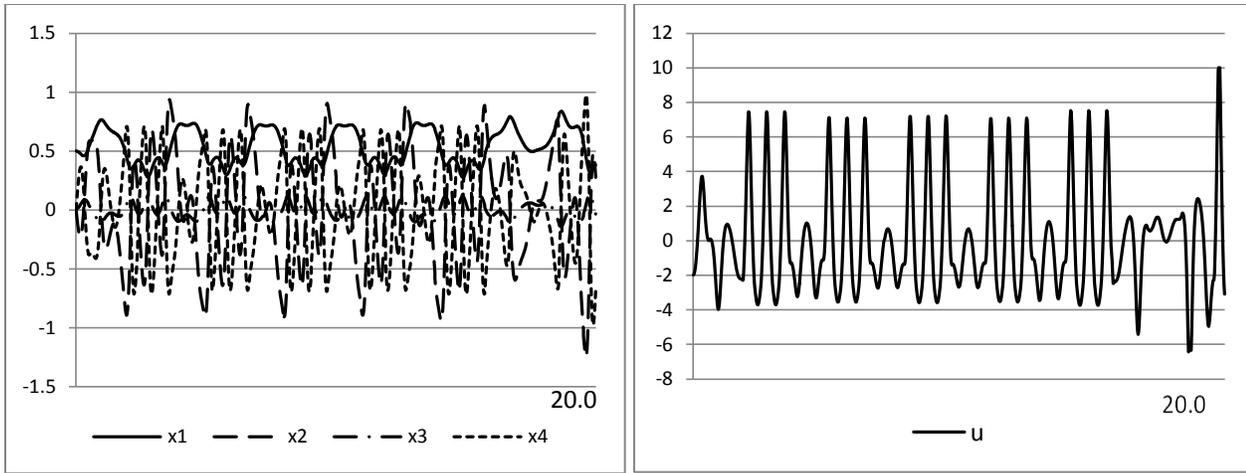


Fig. 24 State variables convergence and controller *L2-13*.

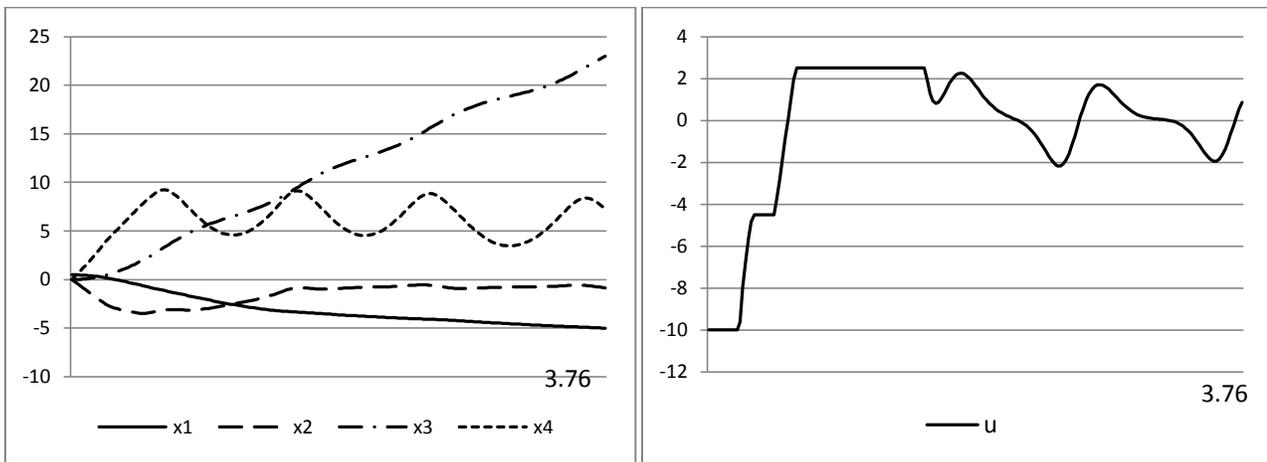


Fig. 25 State variables convergence and controller *L2-12*.

The well performing controller *L3-14-3-2* is analysed and simulation results are shown in Fig. 30—31. The additional rule base in the control system *L3-14-3* produces a better result than smaller rule base in *L3-14*.

In general, the approximate controller u_2 (intermediary control between layer 2 and layer 3) maintained control of the system for longer periods of time than approximate controller u_1 (intermediate control between layer 1 and layer 2), which is not surprising as the controller u_2 has a larger knowledge base to rely on.

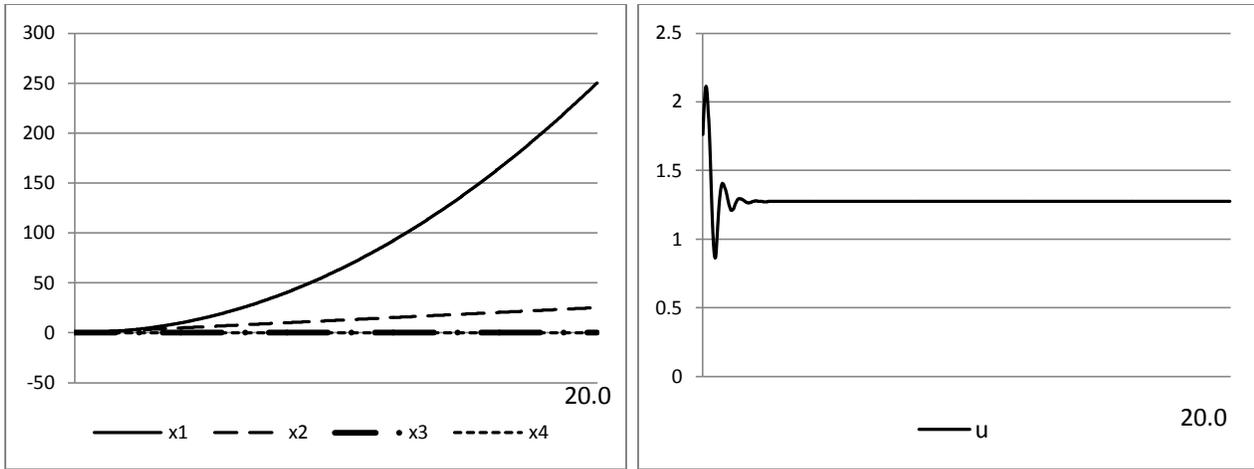


Fig. 26 State variables convergence and controller $L3-34-1$.

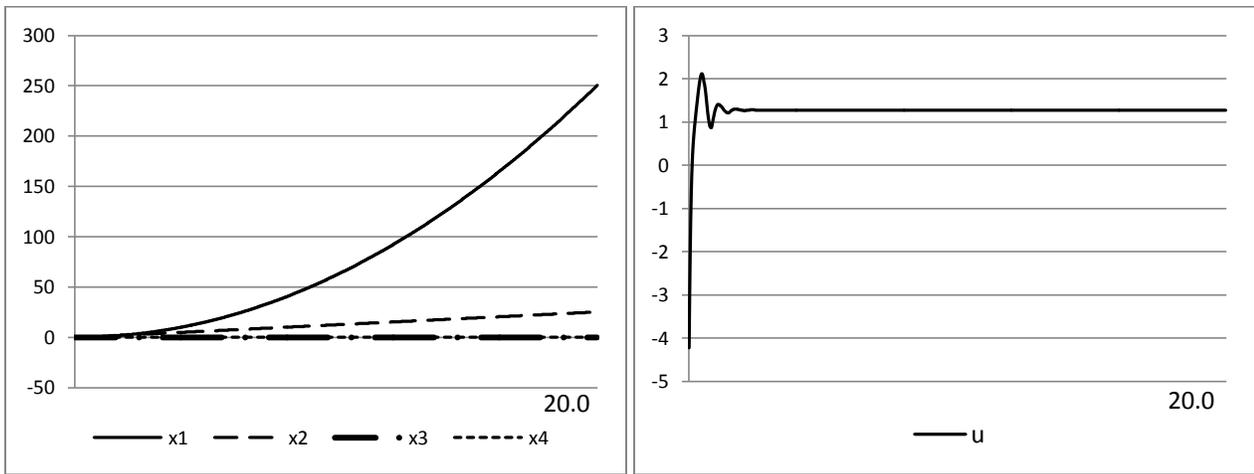


Fig. 27 State variables convergence and controller $L3-34$.

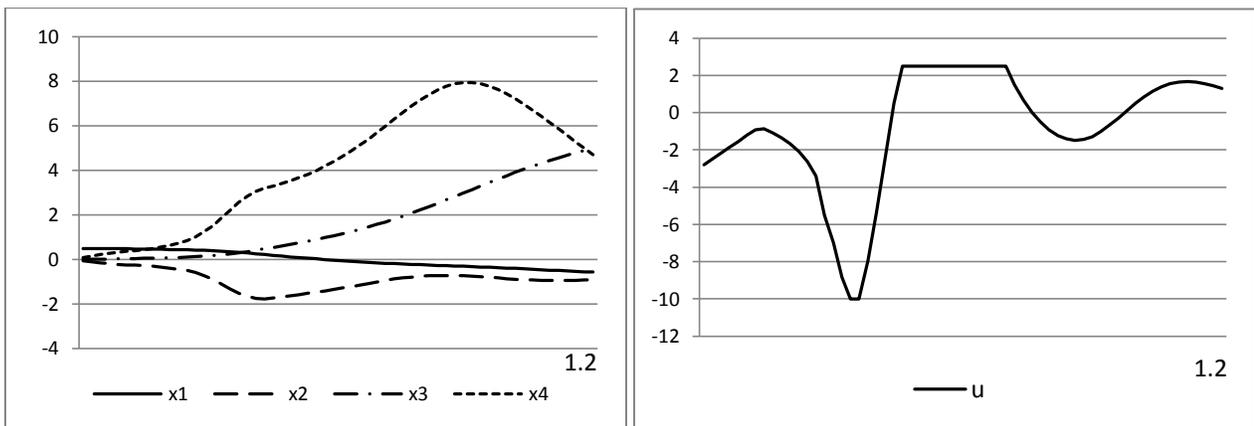


Fig. 28 State variables convergence and controller $L3-23-4$.

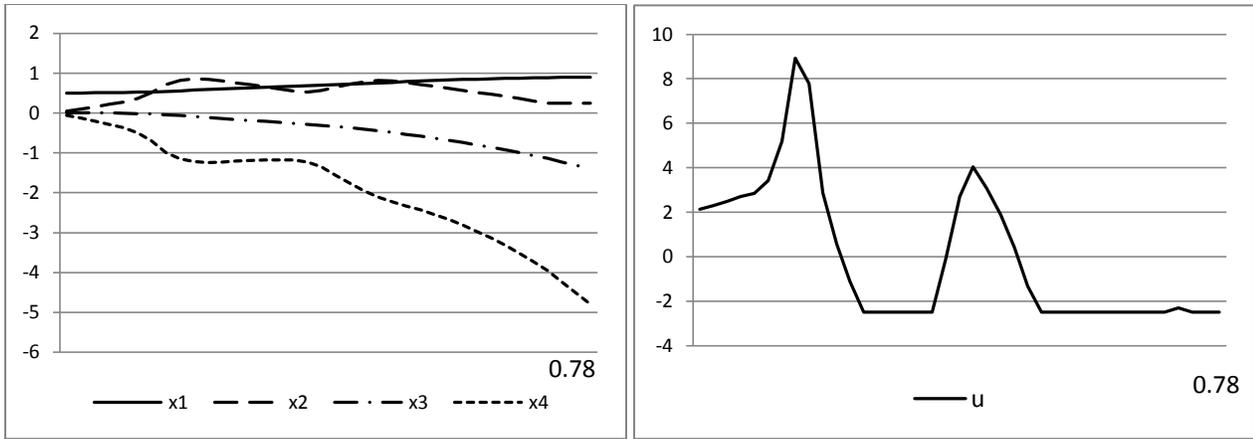


Fig. 29 State variables convergence and controller $L3-23$.

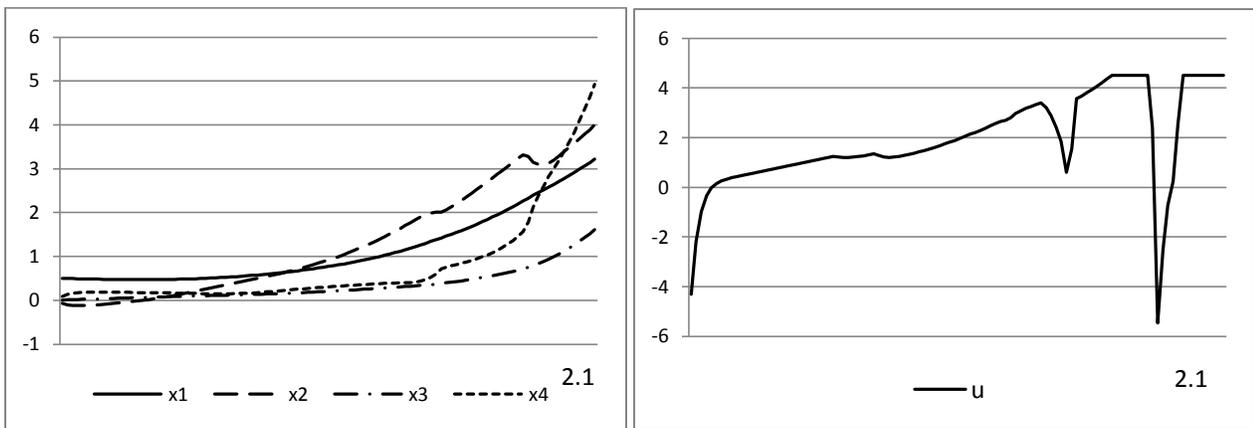


Fig. 30 State variables convergence and controller $L3-14-3$.

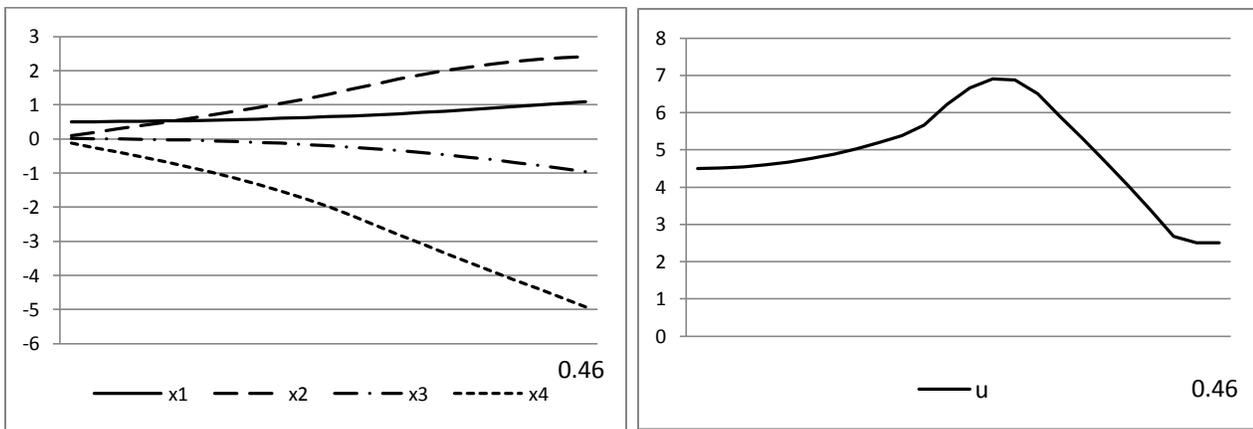


Fig. 31 State variables convergence and controller $L3-14$.

8.3 Controller remarks

The approximate controllers even from the best performing topologies did not exhibit the same quality of control as the final controller u . In some cases, approximate controllers from the ‘worse’ performing topologies, performed reasonably well. The simulations with removed layers clearly demonstrate that the hierarchical structure cannot be assembled from a well-performing component rule-bases but the HFS needs to be developed as a complete topology; hierarchical structure and input configuration.

9 Comparison of results

The developed controllers (see Tables 1, 4, and 5) compare favourably with similar control techniques [41, 47—51]. The fastest stabilization time 1.68 sec was achieved by controller with topology $L4-3-4-2-1$. However, the main aim of this investigation is the impact of HFS topology on controller performance rather than achieving the best controller performance.

Y.J. Mon and C.M. Lin [35] using hierarchical sliding mode control achieved inverted pendulum stabilization in about 10 sec (for initial pole angle $x_3 = \pi/3$). Considering larger pole angle this stabilisation time is a good result. M. Akole and B. Tyagi [59] fuzzy controller was tested for small disturbances and achieved good controller response, better than PID controller, and stabilization in 5-10 sec. One of the fastest stabilization times, about 1-1.5 sec, was achieved by Y. Becerikli and B.K. Celik [60]. Comparison of abovementioned controllers is difficult as the system model, system parameters, and initial conditions were different in each case.

10 Conclusions

In this paper the EA based approach for topology selection was examined. The EA approach enabled comprehensive analysis of simulation results and detailed conclusions are presented below. Fine-tuning of the EA parameters allowed achieving better controller performance for various topologies.

It was observed that the performance of the fuzzy controller is not related to the EA learning speed. It has been shown that it is important to select the correct input variables into the first layer to achieve effective and accurate control. Furthermore, structure of the second and third layer in 3-layered HFS plays a significant role as reversed order in input in those layers produced dramatically different results, as shown for example in case of $L3-13-2-4$ (poor results) and $L3-13-4-2$ (good results). Similarly, controller $L3-14-2-3$ shows poor results and $L3-14-3-2$ good results. Both cases illustrate how intricate interdependencies between input variables can be.

It was shown that the inverted pendulum system should be decomposed into two input variables groupings:

- cart variables: x_1 and x_2 (cart's position and its velocity).
- pole variables: x_3 and x_4 (pole angle and pole's angular velocity).

Results from the 4-layered HFS simulations established that the most influential variable is x_4 (angular velocity), then x_3 (pole angle) in the first grouping followed by x_2 (cart's velocity) and x_1 (cart's position) in the second grouping. Simulation results obtained for two, three and four layered HFS confirm that it is important to control the inverted pendulum, by examining first its angular speed and angular position then the cart's speed and position.

Three-layered topology breaks strong interdependence between state variables in layers 2 and 3 but it does not have adverse effect on the controller performance for topologies $L3-34-1-2$ and $L3-34-2-1$ as this decomposition reflects physical properties of the system (ranking of the most influential variables). For topologies $L3-14-2-3$ or $L3-14-3-2$ the difference in decomposition has a profound effect. Decomposition needs to reflect the physical properties of the system under consideration and it requires grouping of the input variables along weak interdependences between state variables. The inverted pendulum can be decomposed into two subsystems: the cart represented by x_1 and x_2 , and the pole represented by x_3 and x_4 . Swapping the input variables between the layers but preserving to some extent abovementioned groupings has little effect on the controller performance. When this grouping principle is broken, the results are often detrimental (depending which variables are more influential in the dynamical system). The simulation for 4-layered topologies show that the topology $L4-4-3-2-1$ is the most consistent controller in ten different simulations indicating the ranking of the most influential input variables: first - x_4 , second - x_3 , third - x_2 , and finally x_1 .

The initial population (randomly generated in the simulations) has significant impact on the evolution of the knowledge base. Some controllers, from ten control systems developed for each topology, differ considerably in their performance. Therefore a simulation resulting in a single controller should not be regarded as a sufficient representation of controllers developed for any particular topology. Especially, if the EA does not produce a relatively uniform population at the end of the algorithm. Developing a relatively homogenous set of controllers requires careful fine-tuning of the EA parameters and usually a large number of generations.

The controller with topology $L1-1234$ stabilises the system relatively well, with no preference given to any input variable, and interdependence between input variables (being locked in the fuzzy rules) remains hidden. Only by decomposition of the HFS (by breaking a single knowledge base into a

hierarchically structured knowledge base) this interdependence comes into play with dramatic effect.

The topologies analysis shows that the 2-layered HFS provides a slightly better solution to the control problem of the inverted pendulum than 3-layered HFS. This result reflects the physical nature of the inverted pendulum system with pole and cart variables grouped in two separate subsystems that are mirrored in the 2-layered HFS. However, the 3-layered HFS significantly reduces the size of the knowledge bases while providing control system of similar performance.

The investigation into HFS topologies suggests that the size of the knowledge base is not an important factor in controller performance. On one side there is topology *L2-3-412* with 880 fuzzy rules or single layer topology *L1-1234* with 625 fuzzy rules, and on the other side 3-layered topology with 95 fuzzy rules in its knowledge base, except for 3-layered ‘alternative topologies’, with some of them providing similar controller performance. However, the topology of the HFS seems to be the decisive factor in controller performance.

Similarly, the HFS topologies investigation shows that the number of layers is not an important factor in the controller performance (in terms of control magnitude and stabilisation rate of the state variables). This fact allows large single knowledge base to be replaced with the HFS without loss of controller performance. In fact, the HFS produced more efficient controllers (in terms of system stabilisation) than single layer controller except for control magnitude that is lowest for a single layer fuzzy system.

Experiments with the HFS with layers removed show that the HFS is not a mere sum of its rule bases in the component layers. Topology of the HFS is a key factor in the performance of the controller. It has been shown that the HFS needs to be considered in its entirety, not as an assembly of the better or worse performing component layers.

The simulation results indicate that a particular input configuration in the HFS layers is more important than the number of layers as good controller performance was achieved for *L2-34-12*, *L3-34-2-1*, and *L4-3-4-2-1*. This indicates that interdependence of variables plays a crucial role in finding the ‘optimal’ HFS for a particular problem. Examining the nature of variables interdependence is a key to an automated determination of the decomposition of the fuzzy model of control, i.e, selection of the optimal or near-optimal topology. The decomposition of the hierarchical fuzzy structure should be performed along weak interdependency between input variables. However, with more complex dynamical systems there might be multiple weak interdependencies in input configuration. In such cases either expert knowledge is required to resolve the decomposition problem or an automated process that finds optimal or near optimal hierarchical fuzzy topology.

References

1. Cordon O, Herrera F, Hoffmann F, Magdalena L (2001) Genetic fuzzy systems: evolutionary tuning and learning of fuzzy knowledge bases. *Advances in Fuzzy Systems Applications and Theory*, Vol. 19, World Scientific Publishing.
2. Cordon O, Herrera F, Zwir I (2001) Linguistic modeling of hierarchical systems of linguistic rules. *IEEE Transactions on Fuzzy Systems* 10(1): 2–20.
3. Konar A (2005) *Computational intelligence*. Springer Verlag, Berlin.
4. Babuska R (2009) *Computational intelligence in modelling and control*. Delft University of Technology. http://www.dcsc.tudelft.nl/~rbabuska/CTU/transp/lecture_notes_ctu.pdf
5. Zadeh LA (1973) Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics* 3(1): 177–200.
6. Pedrycz W (1984) An identification algorithm in fuzzy relational systems. *Fuzzy Sets and Systems* 13:153–167.
7. Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15(1):116–132.
8. Raju GVS, Zhou J, Kisner RA (1991) Hierarchical fuzzy control. *International Journal on Control*, 54(5):1201–1216.
9. Zajczkowski J, Verma B (2010) An evolutionary algorithm based approach for selection of topologies in hierarchical fuzzy systems. In: *WCCI 2010 IEEE World Congress on Computational Intelligence*, July 18-23, 2010 CCIB, Barcelona, Spain, pp 976–983.
10. Zajczkowski J, Stonier RJ (2008) Analysis of hierarchical control for the inverted pendulum. *Complexity International*, Vol. 12, Paper ID: msid49. <http://www.complexity.org.au/vol12/msid49/>
11. Stonier RJ, Stacey AJ, Messom C (1998) Learning fuzzy controls for the inverted pendulum. In: *Proceedings of ISCA 7th International Conference on Intelligent Systems*, Melun, pp 64–67.
12. Chen CS, Chen WL (1998) Robust adaptive sliding-mode control using fuzzy modelling for an inverted-pendulum system. *IEEE Transactions on Industrial Electronics* 45(2):297–306.
13. Chen BS, Uang HJ, Tseng CS (1999) Robustness design of nonlinear dynamic systems via fuzzy linear control. *IEEE Transactions on Fuzzy Systems* 7(5): 571–585.
14. Zhong W, Rock H (2001) Energy and passivity based control of the double inverted pendulum on a cart. *2001 IEEE Conference on Control Applications*, pp 896–901.
15. Durr P, Mattiussi C (2010) Genetic representation and evolvability of modular neural controllers. *IEEE Computational Intelligence Magazine*, August 2010, pp 11-19. doi: 10.1109/MCI.2010.937319
16. Acampora G (2010) Exploiting Timed Automata-based Fuzzy Controllers and data mining to detect computer network intrusions. In: *WCCI 2010 IEEE World Congress on Computational Intelligence, FUZZ*, July 18-23, 2010, Barcelona, Spain, pp 1381-1388.
17. Acampora G, Loia V, Vitiello A (2010) Hybridizing fuzzy control and timed automata for modeling variable structure fuzzy systems. In: *WCCI 2010 IEEE World Congress on Computational Intelligence, FUZZ*, July 18-23, 2010, Barcelona, Spain, pp 1894-1901.
18. Hsu YC, Chen G, Li HX, (2001) A fuzzy adaptive variable structure controller with applications to robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics* 31(3):331–340.

19. Huang YP, Wang SF (2000) Designing a fuzzy model by adaptive macroevolution genetic algorithms. *Fuzzy Sets and Systems* 113: 367–379.
20. Abraham A (2005) Adaptation of fuzzy inference system using neural learning, *fuzzy system engineering: Theory and practice*. Eds. Edjah, N. *et al.* Springer-Verlag, Chapter 3, pp 53–83.
21. Chen Y, Yang B, Abraham A, Peng L (2007) Automatic design of hierarchical Takagi-Sugeno type fuzzy systems using evolutionary algorithms. *IEEE Transactions on Fuzzy Systems* 15(3):385–397. doi: 10.1016/S0020-0255(01)00140-2
22. Torra V (2002) A review of the construction of hierarchical fuzzy systems. *International Journal of Intelligent Systems* 17(5): 531—543. doi: 10.1002/int.10036
23. Tunstel E, de Oliveira MAA, Berman S (2002) Fuzzy behavior hierarchies for multi-robot control. *International Journal of Intelligent Systems*, 17(5): 449–470. doi: 10.1002/int.10032
24. Magdalena L (2002) On the role of context in hierarchical fuzzy controllers. *International Journal of Intelligent Systems*, 17(5): 471–493. doi: 10.1002/int.10033
25. Tachibana K, Furuhashi T (2002) A structure identification method of submodels for hierarchical fuzzy modeling using the multiple objective genetic algorithm. *International Journal of Intelligent Systems*, 17 (5): 495–513. doi: 10.1002/int.10034
26. Kikuchi H, Takagi N (2002) Hierarchical fuzzy modeling and jointly expandable functions. *International Journal of Intelligent Systems*, 17 (5): 515–529. doi: 10.1002/int.10035
27. Lee ML, Chung HY, Yu FM (2003) Modeling of hierarchical fuzzy systems. *Fuzzy Sets and Systems* 138(2): 343-361. doi: 10.1016/S0165-0114(02)00517-1
28. Cordon O, Herrera F, Zwir I (2002) A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology. *Fuzzy Sets and Systems* 138(2): 307-341. doi:10.1016/j.asoc.2006.12.001
29. Delgado MR, von Zuben F, Gomide F (2003) Hierarchical genetic fuzzy systems. *Fuzzy Sets and Systems*, 138(2): 307-341. doi: 10.1016/S0165-0114(02)00388-3
30. Sushmita S, Chaudhury S (2007) Hierarchical fuzzy case based reasoning with multi-criteria decision making for financial applications. In: *Pattern recognition and machine intelligence. Lecture Notes in Computer Science*, Springer, Volume 4815/2007, 226-234. doi:10.1007/978-3-540-77046-6_28
31. Cheong F (2008) A hierarchical fuzzy system with high input dimensions for forecasting foreign exchange rates. *International Journal of Artificial Intelligence and Soft Computing* 1(1): 15-24.
32. Yager RR. (1998) On the construction of hierarchical fuzzy systems models. *IEEE Transactions on Systems, Man and Cybernetics* 28(1):55-66.
33. Holve R (1997) Rule generation for hierarchical fuzzy systems. In: *Fuzzy Information Processing Society, 1997. NAFIPS '97*. pp 444 - 449. doi: 10.1109/NAFIPS.1997.624082
34. Sindelar R (2005) Hierarchical fuzzy systems. In: *Proceedings of the 16th IFAC World Congress, 2005, Vol 15 (1)*. doi: 10.3182/20050703-6-CZ-1902.01119
35. Mon YJ, Lin CM (2002) Hierarchical fuzzy sliding-mode control. In: *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, pp 656–661. doi: 10.1109/FUZZ.2002.1005070
36. Yeh ZM, Li KH (2004) A systematic approach for designing multistage fuzzy control systems. *Fuzzy Sets and Systems* 143(2): 251–273.

37. Stonier RJ, Mohammadian M (2004) Multi-layered and hierarchical fuzzy modelling using evolutionary algorithms. In: Proceedings of CIMCA '2004, Gold Coast, pp 321—344.
38. Dasgupta D (1998) Evolving neuro-controllers for a dynamic system using structured genetic algorithm. *Applied Intelligence* 8: 113–121.
39. Wang W, Yi J, Zhao D, X. Liu X (2005) Design of cascade fuzzy sliding-mode controller. In: 2005 American Control Conference, Portland, USA, pp 4649–4654.
40. Cheong F, Lai R (2007) Designing a hierarchical fuzzy controller using the differential evolution approach. *Applied Soft Computing* 7(2):481–491.
41. Magdalena L (1998) Hierarchical Fuzzy Control of a Complex System using Meta-knowledge. In: Proceedings of the 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, pp 630—637.
42. Lei S, Langari R (2003) Synthesis and approximation of fuzzy logic controllers for nonlinear systems. *International Journal of Fuzzy Systems* 5(2):98–105.
43. Lin CM, Mon YJ (2005) Decoupling control by hierarchical fuzzy sliding-mode controller. *IEEE Transactions on Control Systems Technology* 13(4): 593—589.
44. Shuliang L, Langari R (2000) Hierarchical fuzzy logic control of a double inverted pendulum. In: Fuzzy System 2000, FUZZ IEEE 2000, The Ninth IEEE International Conference, Vol 2, pp 1074—1077.
45. Castillo O, Cazarez N, Rico D (2006) Intelligent control of dynamic systems using type-2 fuzzy logic and stability issues. *International Mathematical Forum* 1(28): 1371—1382.
46. Raju S, Zhou J (1993) Adaptive hierarchical fuzzy controller. *IEEE Transactions on Systems, Man, Cybernetics* 23(4): 973—980.
47. Yi J, Yubazaki N (2000). Stabilization fuzzy control of inverted pendulum systems. *Artificial Intelligence in Engineering* 14: 153–163.
48. Yu WS, Sun CJ (2001) Fuzzy model based adaptive control for a class of nonlinear systems. *IEEE Transactions on Fuzzy Systems* 9: 413–425.
49. Chang W, Park JB, Joo YH, Chen G (2002) Design of robust fuzzy-model based controller with sliding mode control for SISO nonlinear systems. *Fuzzy Sets and Systems* 125: 1–22.
50. Koo TJ (2001) Stable Model Reference Adaptive Fuzzy Control of a Class of Nonlinear Systems. *IEEE Transactions on Fuzzy Systems* 9(4): 624–636.
51. Qiao F, Zhu QM, Winfield A, Melhuish C (2003) Fuzzy sliding mode control for discrete nonlinear systems. *Transactions of China Automation Society* 22(2): 313–315.
52. Wang LX (1997) A Course in fuzzy systems and control. Prentice Hall, NJ, USA.
53. Khan SA, Engelbrecht AP (2010) A fuzzy particle swarm optimization algorithm for computer communication network topology design. *Applied Intelligence*. doi: 10.1007/s10489-010-0251-2
54. Gacto MJ, Alcalá R, Herrera F (2010) A multi-objective evolutionary algorithm for an effective tuning of a fuzzy logic controllers in heating, ventilating and air conditioning systems. *Applied Intelligence*. doi: 10.1007/s10489-010-0264-x
55. Erus G, Polat F (2007) A layered approach to learning coordination knowledge in multiagent environments. *Applied Intelligence* 27: 249–267. doi: 10.1007/s10489-006-0034-y

56. Chen CM (2005) A hierarchical neural network document classifier with linguistic feature selection. *Applied Intelligence* 23: 277–294.
57. Hong TP, Lin KY, Chien BC (2003) Mining fuzzy multiple-level association rules from quantitative data. *Applied Intelligence* 18: 79–90.
58. Cho SB, Shimohara K (1998) Evolutionary learning of modular neural networks with genetic programming. *Applied Intelligence* 9: 191–200.
59. Akole M, Tyagi B (2008) Design of fuzzy logic controller for nonlinear model of inverted pendulum-cart system. In: XXXII National Systems Conference NSC 2008, pp. 750–755.
60. Beceriklia Y, Celik BK (2007) Fuzzy control of inverted pendulum and concept of stability using Java application. *Mathematical and Computer Modeling*. 46(1-2): 24–37.