

Hierarchical Fuzzy Logic Systems and Controlling Vehicles in Computer Games

Krzysztof Renkas, Adam Niewiadomski

Institute of Information Technology

Lodz University of Technology

ul. Wólczajska 215, 90-924 Łódź, Poland

Krzysztof.Renkas@gmail.com, Adam.Niewiadomski@p.lodz.pl

Abstract. *This paper presents a possible application of fuzzy logic systems to control vehicles in computer games. A new architecture of a fuzzy logic system is here proposed: Hierarchical Fuzzy Controller, that is composed of several fuzzy controllers in their classic meaning. "Hierarchical" means that fuzzy sets produced as output of one of fuzzy controllers are then processed as input of another fuzzy controller. The use of such a controller significantly enhances the possibilities of computational intelligence methods in single-player games, i.e. where the enemy is controlled by an agent simulating real behaviour (movement, decisions, etc.). The proposal of an original architecture of Hierarchical Fuzzy Controller, built with fuzzy controllers (in the sense of Mamdani), and discussing advantages of using this architecture to control military vehicles in a 2D single-player game, in comparison to classic fuzzy controllers, are the main scope of the paper.*

Keywords: *fuzzy logic systems, fuzzy controller, hierarchical fuzzy logic systems, hierarchical fuzzy controller, simulation in computer games, agents in computer games.*

1. Introduction

This paper addresses issues of simulations in computer games. In particular, we are interested in computational intelligence methods based on fuzzy logic

systems that make it possible to simulate an enemy in single-player games. The new solution proposed here is to replace classic fuzzy logic systems (Fuzzy Controllers, FC) with *hierarchical fuzzy logic systems* (Hierarchical Fuzzy Controllers, HFC). The main idea of such a solution is that output (fuzzy or defuzzified) of one of fuzzy controllers are then considered as input of another fuzzy controller in its traditional meaning. The hierarchical structure of such a fuzzy logic system is illustrated in Figure 2 and discussed in details in Section 3.

In general, fuzzy logic systems or fuzzy controllers are useful in case when a controlled process is not linear and the use of traditional controllers may appear inefficient. *Fuzzy Controller (FC)* is a control system based on fuzzy logic [1], which make decisions based on knowledge containing the rules like *IF ... THEN ...*; with unspecified predicates [2]. Fuzzy controlling includes three stages: fuzzification, inference and defuzzification. There are two general models of fuzzy controllers: Mamdani model and Takagi-Sugeno-Kanga model (TSK) [2]. Mamdani model operates on fuzzy antecedents and fuzzy consequents using defuzzification of a fuzzy output to get crisp value while TSK uses a bit different construction of rules to evaluate crisp output [3]. In this paper we refer only to the Mamdani model.

Scientific literature enumerates a huge number of application of fuzzy controlling, in this paper, we concentrate on proposals of architecture of Hierarchical Fuzzy Controllers (HFC). The main difference between HFC and FC is that HFC is built of several traditional FCs, and is one complex inference system. Figure 1 shows examples of three controllers. The first one is a traditional fuzzy controller (A), and the next two are sample structures of a HFC (B, C). As it is shown, in case of B or C, output of one of FCs becomes input of another one in the HFC structure. This structure may contain traditional fuzzy controllers of both Mamdani and TSK types.

In this paper, we propose an original structure of HFC as a helpful tool for controlling vehicles in computer games. Applying this modern structure allows us to significantly decrease the number of rules in the inference block of the system and to simplify the process of modelling and simulating the behaviour of vehicles. On the other hand, using traditional FCs to achieve these effects appeared difficult and time-consuming. The proposal was firstly described in [4]. Experiments are conducted in the base of game entitled *Tank 1990*, in particular, its newer version, *Tank 1990-2012*, developed by the author, see Sections 3.2 and 4.

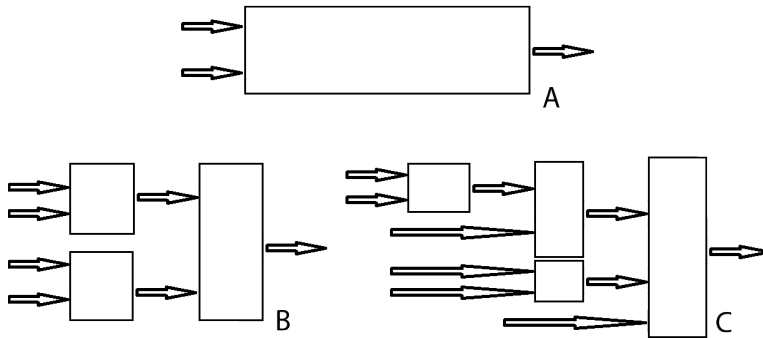


Figure 1. Example of three fuzzy controllers, one simple (A) and two hierarchical (B, C)

The rest of the paper is organized as follows: Section 2 contains some literature references on known hierarchical fuzzy logic systems. The main concepts of architecture of original hierarchical fuzzy controller for game purposes are given in Subsection 3.1 in details. Subsection 3.2 application of our HFC to our specific problem. In Section 4, tests of the designed systems and the results with preliminary discussion are described. The last Section 5 contains conclusions and some future directions of the research.

2. Literature references

The authors of [5] introduces a HFC for agricultural robots, independently navigating in a natural environment for cultivation or irrigation purposes. Another interesting example concerns truck backer-upper system [5, 6]. Controlling smaller cars are mentioned in paper [7] on controlling the velocity and the gap between cars in single lane platoons, tested on highway I-15 in California. Paper [8] presents two case studies about integrated planning and control of mobile robots from point A to point B avoiding obstacles. For example a robot has a knowledge that he should turn left on intersection C, and on intelligent vehicle/highway system that improves efficiency of a highway traffic by grouping cars into platoons. Paper [9] concerns application of a HFC to an urban traffic network with three junctions in rush hours.

3. Hierarchical fuzzy controller and 2D vehicle simulation

In case of vehicle controlling, it is more important to provide an intelligent course of actions to achieve an assumed goal, than to provide human behaviours such as hesitation or wrong decisions. In the context of the applied methods, we suggest some numerical methods to solve the described problems, for example, to determine the shortest path to find optimal route for a vehicle in a given environment. In more complex problems, we indicate the need for application of an artificial intelligence.

3.1. An architecture of hierarchical fuzzy controller

Analysis of different problems solution based on multistage inference we could note that applied controllers usually do not use the hierarchical structure. Outputs of such FCs are returned as independent values, without any connections between them during inference.

In contrast to them, the newly proposed HFC structure combines all of FCs into one system, where the output of one FC is the input of another one. Final output of this system is one crisp value. Furthermore, not every of combined controllers works during each iteration of inference. Figure 2 shows the discussed structure.

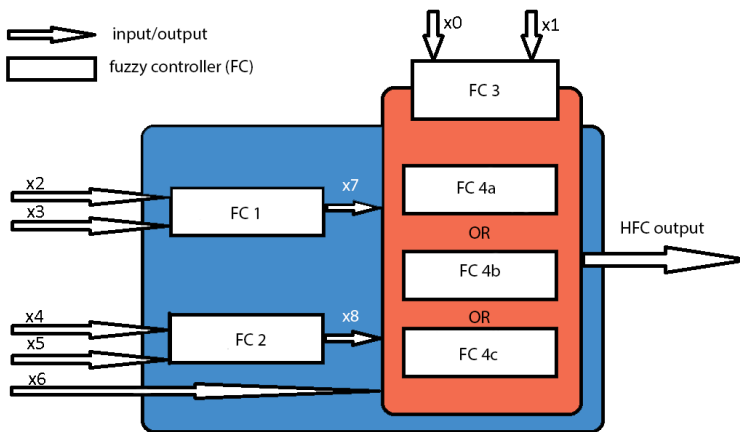


Figure 2. General diagram of designed HFC structure [own elaboration]

We can see different locations of unit controllers, communicating with each other in different ways in Figure 2. Application of this HFC to a specific case is described in Subsection 3.2 in details. To maintain a certain level of abstraction, we can assume that the system has 7 inputs denoted as x_0, \dots, x_6 assigned to different internal drivers, which can additionally takes other outputs as inputs, e.g. x_7 and x_8 . The system could be started many times, for example $m \in \mathbb{N}$ times, i.e. we consider m iterations. In each iteration, system can be started n times for different data sets. For example, controlling a platoon of cars, inclusive n cars includes m iterations that has common elements for each car. In each iteration, one can call the inference for each car (data set), that is to say n times. A FCs on a blue background (see Fig. 2) are run in each iteration for each data set, so this is $m \cdot n$ times. FC on the salmon background, Fig. 2, are some kind of "sub-controller" managed by FC no. 3. For each m iterations FC 3 is running once, making decision which of the FC marked as $4a, \dots, 4c$ will be running for n next data sets in actual iteration.

By using the proposed structure of HFC, we can simplify tank control problem in a computer game. In this game, HFC manages a battalion of tanks. In this case we can treat iteration as inference for the entire battalion. A single set of data concerns a specific tank in this battalion. At the start of iteration, FC 3 selects the strategy on the basis of the game (available strategies: defensive, offensive and mixed). FC $4a, \dots, 4c$ correspond to the mentioned strategies chosen by FC 3. In the next step for each tank, controller no. 1 and 2 are activated, and one of next controller, $4a, 4b$, or $4c$, returns the final system decision about action for a given tank.

3.2. HFC in controlling tanks' activity in computer games

Chooosed problem to solve has been defined as controlling military vehicles during clashes in computer games. General rules of the game were drawn from *Tank 1990*¹ developed by Namco [11].

The main goal of applying HFC based on dedicated structure described in Section 3 is to add artificial intelligence method to the game, in order to increase efficiency of tanks movement and fight during the simulated battle. The final result, mostly important for users who are also players, is a greater difficulty in achieving victory over their opponents. Furthermore, applying this HFC has to show the

¹Basic information about game *Battle City*, an earlier version of the game *Tank 1990*. [10]

simplicity of the proposed solution for controlling tanks by an HFC in comparison to typical FC.

Input data and controller knowledge Input data for inference come from the simulator and player does not operate directly on the HFC inputs. Input data are related to the game and are expressed by the following linguistic variables:

- opponent tanks count (x_0)
- average force of opponent tanks (x_1)
- distance to the nearest opponent tank (x_2)
- force difference between our tank and nearest opponent tank (x_3)
- number of allies - tanks that belong to our battalion (x_4)
- average force of allies (x_5)
- tank is being attacked (x_6)

The fuzzy controllers are based on Mamdani's model with fuzzy antecedents and fuzzy consequents. Examples of rules are listed below.

- strategy controller rules

```
RULE 1 : IF TANKS_COUNT IS SMALL AND AVERAGE_TANKS_FORCE
IS SMALL THEN STRATEGY IS DEFENSIVE;
...
RULE 6 : IF TANKS_COUNT IS BIG AND AVERAGE_TANKS_FORCE
IS MEDIUM THEN STRATEGY IS OFFENSIVE;
...
```

- Support controller rules

```
RULE 1 : IF ALLIES_COUNT IS SMALL AND
AVERAGE_ALLIES_FORCE IS SMALL THEN SUPPORT IS ZERO;
...
RULE 6 : IF ALLIES_COUNT IS BIG AND AVERAGE_ALLIES_FORCE
IS MEDIUM THEN SUPPORT IS BIG;
...
```

- Action controller rules (offensive strategy)

```

RULE 0 : IF TANK_IS_BEING_ATTACKED IS YES THEN ACTION
IS ATTACK;
RULE 1 : IF RISK IS ZERO AND SUPPORT IS ZERO
THEN ACTION IS PATROL;
RULE 2 : IF RISK IS SMALL AND SUPPORT IS ZERO
THEN ACTION IS WAIT;
...

```

The simulator and the use of HFC Game *Tank 1990-2012* was created as a simulator for testing and demonstrating designed controller and created rule base. This simulator has been implemented in Java. Figure 3 shows two screenshots of this game.

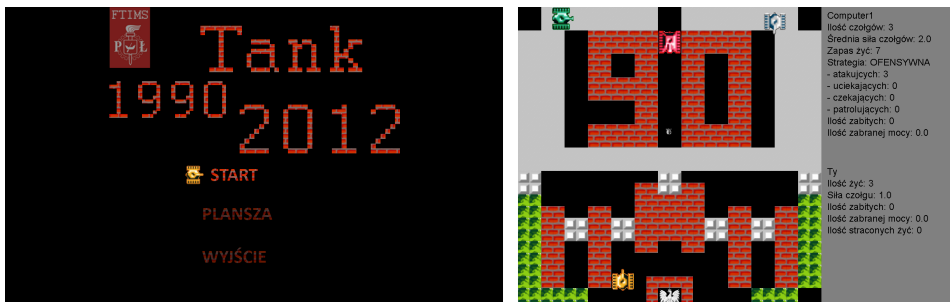


Figure 3. Screenshots of game *Tank 1990-2012*. Home page (on the left) and first stage during battle (on the right). [own elaboration]

Structure of used HFC is shown in Figure 2. Inputs of this HFC are described in section 3.2. FCs 1 and 2 corresponds to the controllers computing the level of risk x_7 and support x_8 . These variables are transferred as an input to the tank action controller $4a, \dots, 4c$. Only one of these FCs is activated in the current iteration depending on the strategy chosen by the FC 3. Offensive, defensive and mixed strategies are allowed. The controlling system selects actions for each tank in battalion. Final decisions allow to escape, attack, stay at the current position and to patrol immediate surroundings.

4. Tests and results

”Tank 1990-2012”: a newer version of classics. Tests The experiment is conducted on the base of the computer game entitled *Tank 1990-2012* created by the author, and it is a newer version of classic *Tank 1990*. Especially, the main difference is the use of a Hierarchical Fuzzy Controller, and the rule base accessible to everyone, so it allows to do own tests².

Additionally in this paper two similar tests were carried out on original game *Tank 1990*. In the first test game *Tank 1990* was ran 72 times, counting losts in subsequent stages, taking stage 11 as the last one. In the next test game was lunched 50 times without doing any moves with tank, counting the total time needed to get the victory by opponents. The average time was compared to the time obtained by the best configuration of our HFC. Test conditions were not the same, for example the speed of vehicles in original game was much more faster.

Also additional summary of HFC rule base has been made in comparison to estimated FC rule base needed to do the same task. This comparison shows simpler and more efficient solution.

Results Table 1 shows results of the first test described in the paragraph above. As it is shown in Table 1 it is quite easy to win playing *Tank 1990* (43 losses, 59.7%). On the other hand, this is almost impossible to win in newer *Tank 1990-2012* (100.0% losses); this is because the latter version of the game is equipped with much more intelligent method of controlling tanks.

Table 1: Information about losses during played games *Tank 1990-2012* and *Tank 1990*.

Results		
stage number	lost (<i>Tank 1990-2012</i>)	lost (<i>Tank 1990</i>)
1	3	1
2	10	4

²The thesis [4] contains three types of tests. The first one applies to general behaviour of game and tanks controlled by HFC. In this test case game was lunched 72 times, counting losses in subsequent stages. The second test is more detailed comparing efficiency of different controller configurations. The game was lunched 50 times for each configuration and the best configuration is selected. The last test concern subjective ”playability”, assessed by people in different ages.

Results		
3	8	2
4	12	5
5	11	4
6	8	4
7	9	6
8	6	5
9	4	3
10	1	4
11	no data	5
lost	72 (100%)	43 (59.2%)
victory	0 (0%)	29 (20.3%)

The second test concern time periods needed to get a victory. Different configurations of HFC are tested and the best are minimum *t-norm*, Łukasiewicz *s-norm* and the middle of maximum (MOM) as defuzzification method. *Tank 1990* was tested in *NES* console simulator. Average results are shown in the Table 2. As we can see, almost identical results are achieved, and the difference is only 0.4 seconds [4].

Table 2: Average times necessary to win in *Tank 1990-2012* and *Tank 1990*.

Averaged results [s]	
<i>Tank 1990-2012</i>	<i>Tank 1990</i>
29.8	29.4

Summarizing HFC rule base, we count $7 + 2 = 9$ linguistic variables (7 as input and 2 as auxiliary). Counting created rules we get $9 + 9 + 9 + 3 \cdot 17 = 78$. During the inference process, 9 rules decide about strategy, 9 rules about support, 9 rules about risk, and 17 rules about the tank's action (each of strategy use different set of rules to select tank action, so that's why we have $3 \cdot 17 = 51$ rules). For example to do one iteration of inference for battalion of 5 tanks, we must select strategy (9 rules) at the start of iteration, then for each tank we must get risk and support value ($9 + 9 = 18$ rules) and at the end choose the final tank action (17 rules). So to do this task we must fire $9 + 5 \cdot (18 + 17) = 184$ rules. Complexity of these rules is 3: two antecedents and one consequent.

To solve this problem using single layered FC, we would have to include all combinations of those 7 input variables into this FC. In this case we need to create $9 \cdot 9 \cdot 9 \cdot 2 = 1458$ rules (strategy, support, risk and info that tank is being attacked). Using this FC to our sample battalion we need to fire $5 \cdot 1458 = 7290$ rules with 7 antecedents. Sample forecasts are presented in Table 3.

Table 3: Number of fired rules needed to infer for four battalions of 1, 5, 10 and 20 tanks in HFC (real values based on the created rule base) and FC (estimated values).

Number of rules		
number of tanks in battalion	HFC	FC
1	44	1 458
5	184	7 290
10	359	14 580
20	709	29 160

The summary of this comparison is depicted by the graph shown in Figure 4.

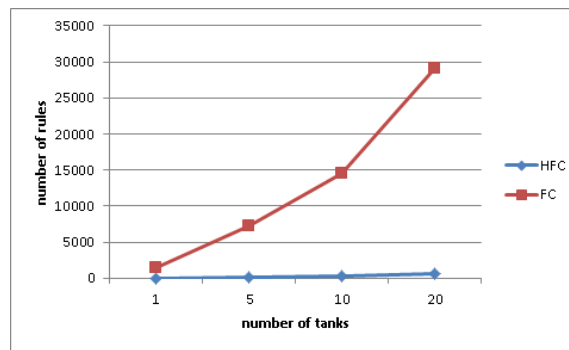


Figure 4. Number of fired rules during one inference for battalion of 1, 5, 10 and 20 tanks. Real HFC rule base and estimated FC rule base are presented.

5. Conclusions and future work

The tests run prove some positive effects of applying Hierarchical Fuzzy Controllers (HFC) to control virtual vehicles in games. The results of the first test,

presented in Table 1, determine much higher game difficulty, i.e. more intelligent behaviour of enemy tanks, controlled by HFC, with respect to the *Tank 1990*. The results of the second test, see Table 2, inform us that the more time-consuming computations (mostly inferences) in case of HFC does not decrease the speed of the game; the lack of negative effect is a really important observation here, mostly because of higher computational complexity of HFC. We also note that use of HFC has a positive impact on the tank control in newer version of the tested game, i.e. *Tank 1990-2012*. The vehicles move intelligently and try to achieve clearly defined goals, while tanks in the older version, i.e. *Tank 1990*, move chaotically and make irrational decisions.

From the point of view of a programmer/developer, HFC make it possible to adapt easily controllers to specific problems appearing in software, not necessarily in games. The modular and hierarchical design of such a controller makes problem easy to understand and allows to find a simple solution to this problem. For instance, the HFC described in Section 3 requires only 78 unique inference rules, while the analogous FC for the same purposes requires 1 458 (!). Furthermore, the rules of FC are more complex, because of using 7 linguistic variables, while the HFC uses only 2.

Our future research may direct towards combining genetic and/or evolutionary algorithms or artificial neural networks with hierarchical fuzzy controllers. Genetic and evolutionary algorithms, as optimisation methods, can be used to find optimal route to the specified destination or object. Neural networks can be useful in dynamic tuning knowledge bases of HFCs.

References

- [1] Zadeh, L. A., *The Concept of a Linguistic Variable and its Applications to Approximate Reasoning (I)*, Information Sciences 8, 1975, pp. 199–249.
- [2] Yager, R. R. and Filev, D. P., *Essentials of Fuzzy Modeling and Control*, A Wiley-Interscience publication, John Wiley & Sons, 1994.
- [3] Kaur, A. and Kaur, A., *Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System*, International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, may 2012.

- [4] Renkas, K., *Hierarchical fuzzy system and controlling military vehicles in computer games*, Master's thesis, Lodz University of Technology, Stefana Żeromskiego 116, Łódź, Poland, september 2012.
- [5] Hagrass, H., Callaghan, V., Colley, M., and Carr-West, M., *A behaviour based hierarchical fuzzy control architecture for agricultural autonomous mobile robots*, Journal of Autonomous Robots, Vol. 13, july 2002, pp. 37–52.
- [6] Riid, A. and Rüstern, E., *Fuzzy hierarchical control of truck and trailer*, BEC 2002: proceedings of the 8th Biennial Baltic Electronics Conference, october 6-9, 2002, pp. 141–144.
- [7] Kim, H. M., Dickerson, J., and Kosko, B., *Fuzzy throttle and brake control for platoons of smart cars*, Fuzzy Sets and Systems, Vol. 84, 1996, pp. 209–234.
- [8] Wang, L.-X., *Modeling and Control of Hierarchical Systems with Fuzzy Systems*, Automatica, Vol. 33, No. 6, 1997, pp. 1041–1053.
- [9] Gegov, A. E. and Frank, P. M., *Hierarchical fuzzy control of multivariable systems*, Fuzzy Sets and Systems, Vol. 72, 1995, pp. 299–310.
- [10] Game description *Battle City*, Available at: http://pl.wikipedia.org/wiki/Battle_City, [access: 6 XI 2011].
- [11] American web page of *Namco*, Available at: <http://www.namcoamerica.com>, [access: 22 VIII 2012].