Design and Implementation of Adaptive Neuro-fuzzy Exergame Controller

Kiavash Fathi, Afrooz Laghaei, Maryam Zekri

*Department of Electrical and Computer Engineering*

*Isfahan University of Technology*

Isfahan, Iran

kiavash.fathi@ec.iut.ac.ir, afrooz.laghaei@ec.iut.ac.ir, mzekri@cc.iut.ac.ir

Javad Rasti

*Faculty of Engineering*

*University of Isfahan*

*University of Isfahan Center of video games*

Isfahan, Iran

rasti@eng.ui.ac.ir

*Abstract*—Video games are among the popular leisure activities among the youth. Due to the long hours of playing video games, many health-related issued have occurred. One practical solution for dealing with the aforementioned problem is to come up with game controllers which promote physical exercises. In this paper an adaptive neuro-fuzzy exergame controller is designed and implemented. The controller is made of four pads and a microcontroller, which sends commands to a game running on a computer. The received signal from the controller is sent to a fine-tuned fuzzy logic system (FLS). The calculated output of the previously trained FLS is one of the defined classes of “ignore”, “press” and “hold”, which is sent as a command to the game running on the computer. The FLS is implemented in such a way that the integration of the controller hardware and the FLS provide an acceptable playing experience for the user. All in all, the implemented system is not a pure FLS, due to the fact that, the users of the controller vary in age and physical characteristics and thus a previously tuned FLS may not provide an acceptable user experience, thus the parameters of the system have to be fine-tuned to best suit the needs of the user. Hence, the user is first asked to press and hold the pads couple of times in order to gather a training data set for fine-tuning the proposed system.

Keywords— Neuro-fuzzy Logic System, Adaptive System, Exergames, Game Controller.

# Introduction

Video games have taken over most of the leisure time of the youth, causing laziness and sedentary [1]. Therefore, it is vital to address this problem by providing novel technologies for developing exergames accessible for a wide range of users. Exergame as a video game which also includes physical activity, can simply solve some health-related problems induced by long hours of playing video games [2]. Additionally, Exergames have the potential to help people of different ages and interests, and make them to get enough physical exercise for maintaining their health [3], [4]. An inseparable part of exergames is the applied controller. The provided input device range from pads which contain pressure activated switches, to more complicated and expensive accessories such as Kinect and PlayStation Move [5]. There is a long history of different ideas used for controlling exergames [6]. In 1983, Autodesk developed *HighCycle*, a cycle bike that the user had to pedal, in order to fly over a virtual landscape. Later in 1998, Konami released *Dance Dance Revolution*. The controller used in this game takes actions from the user through its pressure activated switches [7]. The aforementioned examples are some simple yet novel game controllers. A more complex input device is Kinect, launched in 2010, which can be considered as one of the most significant improvements in detecting players’ motion. Infrared projector, camera and a microchip for determining the location of a nearby object in 3 dimensions, are what Kinect’s system is consisted of. Recently, virtual reality has proved to have the potential to be used in not only entertainment purposes like exergaming but also in training professional athletes [8], [9]. All these examples suggest the importance of developing new devices for exergaming. Exergaming can be also used in public places (e.g. metro stations, malls, etc.) for promoting public health due to the fact that exergames tend to be interesting for people of different ages and interests [10].

In this paper an adaptive and effective controller for exergaming is designed and implemented. The proposed system is consisted of four pads, which the player can use for interacting with computer. By using the aforementioned pads, the user can send commands from the controller to the game the user is playing. The proposed method does not require any additional peripherals like camera or sensors of any kind which is considered as an asset, since such a design can increase the costs. Also the controller can be used in harsh environments, because it only relies on the signals received form the pads and no other factors like lighting, temperature, etc., can affect the performance of the proposed system. Moreover, the applied neuro-fuzzy system is capable of providing the best user experience according to the behavior of the user and also the requirements of the game. When a player is using a keyboard, the task of distinguishing between pressing (tapping once) and holding (holding down a specific key), which is similar to short and long touch in touch screens is easy. But for a game controller of a exergame, which may be used by players of different physical characteristics, the task of analyzing behavior and classifying the action of the user can be more complicated. It can be said that different players need different mappings from the received signal from pads to a command that will be sent to the game which is running on a main computer. The aforementioned mapping is done using different member functions. By changing the parameters of member functions it is possible to adjust the controller to best suit the physical characteristics of the user. Additionally, by applying a fuzzy logic system (FLS), it is possible to use the available domain knowledge for the initial values of the parameters of the member functions. The available knowledge will ease the process of fine-tuning of the game controller. One important constraint to remember is the fact that it is not possible to gather extensive data from the user for fine-tuning, thus it is logical to say that having reasonable initial values for parameters is crucial. Hence, it can be claimed that, an adaptive fuzzy system for game controller is unavoidable. A crisp and invariant approach for classifying the actions of the user can lead to discontent of the user due to the fact that, the user cannot communicate effectively with the system and the user’s playing experience will deteriorate even if the game itself is perfectly designed and guarantees an acceptable playing experience. In the proposed method, this problem is solved by calibrating the controller using the gathered signal from the player. The parameters of the member functions of the applied FLS are fine-tuned using the gathered data from the player, which guarantees the acceptable performance of the controller. On the other hand, it is not practical to develop special games for a game controller because such an approach can limit the options of the user and also makes the game controller applicable for a specific game, even though a simple manipulation and data analysis in the data received from the controller is enough to calibrate the game controller for different games. Having an adaptive system can help in promoting exergames, given the fact that an adaptive game controller does not rely on a specific pattern of data stream for its previously designed games and can be used in many games effectively. By calibrating the game controller, it is possible to adjust it in such a way that the controller will respond effectively and correctly to the user’s actions. In the proposed system, the user is asked to press the pads and hold them for a few times so that the system can be trained according to the physical characteristics of the user. It is logical to say that the time interval of a pad press for a child is much shorter than an adult. The gathered data is then used to change the parameters of a previously designed FLS, which are set in such a way that the controller is likely to provide a satisfying game experience for its users by using the available domain knowledge. The aforementioned system is programmed in MATLAB and the hardware of the system is designed using Autodesk Eagle and Altium designer.

|  |
| --- |
|  |
| Fig. 1. Schematic of the proposed circuit for a pad |

The rest of the paper is as follows. First some preliminaries are introduced and afterwards the implemented controller hardware is explained. Then the main structure of the proposed FLS is introduced, thereafter the process of fine-tuning the FLS is pointed out. Lastly, the integration of the proposed hardware and the FLS is explained.

# Preliminaries

## Fuzzy system

In order to best use the available domain knowledge or data from an expert, a fuzzy system can be implemented. The heart of a fuzzy system is a knowledge base which is made of fuzzy rules. These rules are fuzzy IF-THEN statements which determine the overall behavior of the fuzzy system. Afterwards the fuzzy inference engine is used to acquire a mapping from the input space to output space . The input (and sometimes the output) of a fuzzy system are fuzzy sets with their member functions. In short it can be said that a fuzzy system provides a nonlinear mapping from the knowledge base.

A fuzzy set with a continuous member function is written as

(1)

The integral sign is a collection of all the points with member function . A member function is essentially a defuzzification of a fuzzy description. Member function is determined using the available domain knowledge or from an expert. In the proposed method the structures of the member functions are specified in such a way that the system is applicable to a majority of users. Here, the proposed method uses the gathered data from the user for fine tuning the parameters of the member functions when the controller is being calibrated.

Reasoning in a fuzzy system is to obtain new propositions from the existing ones. Instead of having classical two-valued logic, a fuzzy system allows the value of a proposition to be any value in the interval , which eases the process of approximate reasoning. Propositions are represented by fuzzy sets and for acquiring approximate reasoning some fundamental principles in fuzzy logic were introduced. In what follows Generalized Modus Ponens is explained.

Generalized Modus Ponens [11] states that the given two fuzzy propositions and , a new fuzzy proposition , should be inferred in such a way that, the closer to the closer to . Given fuzzy set and fuzzy relation in which represent the premise , a fuzzy set is inferred as follows.

(2)

In which represents t-norm. Fuzzy inference is essentially the formulation of a given input to an output using fuzzy logic. The proposed method uses Sugeno-type inference. In this method the output of the system uses singleton output membership functions which are either a constant value or a linear function of input values. The defuzzification process is computationally efficient because it uses a weighted average or weighted sum of a few data points [12]. The output of the system can be formulated as

(3)

Where is the number of rules, is the rule firing strength and finally is the rule output level. As it was previously pointed out, can be a constant value or a linear function of the input values [13], [14], [[15](https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html)].

## Adaptive neuro-fuzzy inference system

As it was previously pointed out, the designed system is used by different players with various physical characteristics. Hence, it is vital to design a system which changes according to the demands of its current user. By applying adaptive neuro-fuzzy inference system it is possible to change the parameters of the member functions of the system. The parameters of the member functions change during the training stage, where a training dataset is used to train the available model. Collecting training dataset and preprocessing of the gathered data will be explained in Section V. The parameters are updated using backpropagation. A method, in which the calculate error (sum of the difference between actual and desired outputs) of the system is used to update the parameters of the system. In fact, the calculated gradient vector provides a measurement of how well the system is modelling the input/output data for a set of parameters. Here, the ANFIS from Fuzzy logic toolbox of MATLAB is used for fine-tuning the parameters of the proposed FLS [16].

|  |
| --- |
|  |
| Fig. 2. Adaptive neuro fuzzy game controller’s pad |

# Proposed Controller

## Hardware of the controller

The proposed controller is made of four pads connected to an ATMega32 microcontroller. Other peripherals of the implemented system include: four resistors and four push buttons. The schematic of the proposed circuit for a single pad

|  |
| --- |
|  |
| Fig. 3. Member functions of *Very short*, *Short* and *Long* |

can be seen in Fig. 1. When a user pushes a pad (the pushbutton in the pad) the input signal of the microcontroller changes from high to low. Afterwards if the user releases the pad, the input to the microcontroller changes from low to high. The created rising and falling edges are afterwards used in the implemented FLS to make decisions and predict the desired command the user intended to send to the game which is running on the main computer. One of the pads of the final implemented controller (without the top cover) can be seen in Fig. 2.

## Implemed FLS

In order to improve the user experience of the designed controller, we use fuzzy logic as an adaptive and robust solution for manipulating and processing the received data from the controller. The controller can be used to simulate pressing and also holding a key in keyboard (similar to short and long touch in a touch screen). The task of classification of the received input to an action is done by the designed (and afterwards fine-tuned) FLS. The input of the FLS are fuzzy sets. According to the time that the user has activated a pad, three member functions have been used to cover all the possible scenarios. The output of the controller is the input of the FLS. And the output of the FLS is the command that will be sent to the game from the main computer.

The received input from the user are categorized using the following fuzzy sets:

1) *Very short*: This scenario happens, when the system is exposed to noise and the output of the controller changes due to the noise. Naturally, the time interval of the controller output signal changing from high to low and again from low to high, is very short and this feature of the controller output signal can be used to filter out the effects of the contaminated noise in the input signal.

2) *Short*: When the user only pushes the pads of the controller for a short time for sending a press action, the time interval of the controller output signal changing from high to low and again from low to high, is longer than the previous scenario but not as long as the next scenario (*Hold*).

3) *Long*: Lastly, when the user is trying to send a constant hold action, the output signal will be held at low for a longer time than the previous two scenarios.

The most important challenge in this part of designing the FLS is to find the appropriate member functions for the input signal from the controller. The member functions determine the lower and upper bounds for each state and also the transition between different states. Using the available domain knowledge and by running a few sample test it is decided to use sigmoid function for *Very short* and *Long* fuzzy sets and a variation of Gaussian member function for *Short* fuzzy set. Sigmoid and Gaussian member function are defined as follows.

(4)

(5)

The applied Gaussian member function uses two sets of parameters; one set for the shape of the right-most curve and one for the left-most curve. The member functions of Very short, Short, and Long as a function of time, before fine-tuning can be seen in Fig. 3.

As it was previously discussed, in the proposed method, a zero-order Sugeno fuzzy model is used. The applied rules are as follows:

The applied zero-order Sugeno fuzzy model returns for , for and for . The applied system must find a nonlinear mapping from aforementioned input member functions to these values. Afterwards the output of the fuzzy system is used to send a command to the computer. Given the rules above, the output of the FLS can be seen in Fig. 4.

|  |
| --- |
|  |
| Fig. 4. Output of the fuzzy logic system |

|  |
| --- |
|  |
| Fig. 5. Gathered data from the user |

The initial values of the parameters of the member functions are acquired using some test runs. The most important factor to consider while designing the FLS is the effects of the input member functions on the user experience of the controller. By adjusting the parameters of the input member functions, it can be guaranteed that the user can communicate effectively with the controller and computer. Essentially, the rate of changes in the input member functions greatly affect the user experience. As an example, the member function parameters for an adult have a much slower rate of changes than the parameters set for a child, due to the physical characteristics of the aforementioned users. Thus, it can be concluded that, it is best to change the parameters of the input member functions according to the current user of the controller. The proposed approach for handling this problem is using adaptive neuro-fuzzy systems.

# Training adaptive neuro-fuzzy system

The member function parameters will determine the user experience. The most practical way to adjust the parameters of the FLS is to gather some data from the user and afterwards, change the parameters of the input member functions accordingly. In what follows, this procedure is explained in detail. At first, the user is asked to press different pads on the controller. These data are stored in a dataset. Along these data, a tag is used to determine the action the user took. In the proposed method, in the dataset, stands for a press action. Afterwards, the user is asked to hold different pads. For a hold action the tag is used and is saved in the dataset as the user is interacting with the controller. An example of the gathered data from the user is depicted in Fig. 5. The upper bound for press data and the lower band on the hold data play a significant role in the determination of the effectiveness of the gathered data from the user. It’s best to gather more data from the user when these two bounds are very close. By gathering more data, it is possible to remove conflicting data points and improve the quality of the acquired dataset.

|  |
| --- |
|  |
| Fig. 7. Output of the fuzzy logic system to the testing dataset |

After gathering data from the user, the available data can be used to fine-tune the parameters of the previously designed FLS. ANFIS from MATLAB’s fuzzy logic toolbox is used for fine-tuning. During the fine-tuning only the parameters of the input member functions change and no further changes are made in the FLS. During this process, an error tolerance along with total number of epochs must be determined. It is decided to set the error tolerance to 0.05 and to train the available FLS for 9 epochs. Parameters are updated using backpropagation. After training the FLS with the available dataset, the system’s error will be about 0.03, which is acceptable considering the limited data available in the training dataset. The error plot of the system can be seen in Fig. 6. One vital step in fine-tuning of the system is to check whether the system is overfitted. Limited number of available data points may result in overfitting of the model.

During gathering data some points can be saved to be used later in the test step. Additionally, some data points from the training stage can be used as well to expand the test dataset. The result of testing the fine-tuned FLS can be seen in Fig. 7. The blue dots represent data points from the testing data set and the red asterisks are the outputs of the fine-tuned FLS. It can be clearly seen that all but one data (data point number four) in the test data set are correctly estimated by the fuzzy logic system and thus the blue dots cannot be seen clearly in Fig. 7. Given the fact that the system has an error rate of about 0.03, it seems logical to round the output of the FLS to guarantee the correct output from the FLS. During the test stage it can be seen that, after rounding the output of the FLS, all outputs of the system are correct. Given the acceptable performance of the FLS, it can be claimed that the FLS is not overfitted and the training stage is successful.

# Integrating FLS and the hardware of the Gamecontroller

|  |
| --- |
|  |
| Fig. 6. Error of the fuzzy logic system |
|  |
|  |
|  |
|  |

In this section the trained FLS is used to manipulate the received data from the controller. As it was previously pointed out, raw data from the controller may not provide an acceptable user experience and hence the received data from the controller are manipulated using the trained FLS. The FLS will afterwards send an appropriate command to the game which is running on the main computer, by analyzing the interaction between the user and the controller. The implemented code works as follows. The main computer is reading data from a connected microcontroller. Any changes in the value of the received data is analyzed by the trained FLS and the corresponding command is sent from the trained FLS to the game running on the main computer. These commands include pressing a key on the keyboard, holding a key and the idle mode. As the user is interacting with the controller and is pressing a pad, the received signal from the controller changes from high to low. By receiving the falling edge from the controller, a counter starts counting to store the time that the user has held the pad. While the user is holding the pad, the value of the counter is sent to the FLS. By reaching the lower bound for the hold action, the FLS will send a hold command to the game when the firing power of the input member functions results in a hold action. Of course the sent command changes as the held pad changes. Another scenario happens when the user is trying to press a pad. In this scenario, at first a falling edge is detected in the received signal and after a rather short time, a rising edge is detected. This happens when the user presses and releases a pad to send a press command to the game. When the rising edge is detected in the output signal of the controller, the value of the counter is sent to the FLS and afterwards the value of counter is set to zero. The FLS will then determine whether the intended action was pressing a pad or not. When no pads are pressed or held, there may be some changes in the received values from the controller and considering the fact that by implementing a FLS, the implemented system will be robust, it can be guaranteed that the effects of the noise on the output of the controller will not affect the performance of the system. The pseudocode of implemented system is as follows:

|  |
| --- |
|  |

|  |
| --- |
|  |
| Fig. 8. GUI for configuring and calibrating game controller |

The designed system can be used for playing different video games as long as the games can be controlled using four different commands. Platform games are an excellent example of games which can be easily controlled using the proposed controller. Similarly, maze arcade game such as Ms. Pac-Man can also be played using the proposed method. The only remaining step before applying the proposed controller is to configure the controller and send a list of required keys for different commands in the game to the controller. The game controller is designed in such a way that, it uses the provided list from the user, for controlling the game and sends different commands to the game accordingly. As an example, for playing Ms. Pac-Man, four arrow keys are the only required keys used in playing the game. After configuring the controller, as the user presses or holds a pad, the corresponding command will be sent to the game running on the main computer. All keyboard keys are supported and by using the provided GUI, the user can easily configure and fine-tune the controller (Fig. 8.).

|  |
| --- |
|  |
|  |

As it can be seen in Fig. 8., the user can easily determine the command each pad sends to the game. The special keys’ list contains keys such as space, enter and arrow keys which are frequently used in games of different genres. Additionally, the user can check the current state of the game controller using the label on the bottom side of the GUI. As an example, as it can be seen in Fig .8., the controller now is ready to send commands to the game. Also, the controller can be calibrated using the provided button on the bottom left side. By pressing the aforementioned button, the formerly explained process of fine-tuning starts. The proposed controller is connected to the main computer through a USB port, and as the controller’s software is running the connected port will be occupied. Thus it is important to provide *Run* and *Disconnect* buttons for managing the USB port of the main computer. A fully functioning game controller and the main computer running Ms. Pac-Man, can be seen in Fig. 9.

|  |
| --- |
|  |
| Fig. 9. Game controller and the main computer |

# Conclusion

In this paper an adaptive neuro-fuzzy system was introduced which is integrated into an exergame controller. The controller is connected to a computer and the user can communicate with the game running on the main computer using the provided pads in the controller. The user can simulate pressing or holding a key on the keyboard by pressing or holding the aforementioned pads. The proposed controller can be used in exergaming which can help improve the health condition of its user. In order to improve the user experience of the controller, it has been decided to use a FLS for classifying the actions of the user. The predefined FLS’s parameters are set by running some test runs and coming up with a model which is likely to provide an acceptable user experience. But given the fact that the controller is used by different people of different ages and physical characteristics, it is best to change the parameters of the FLS. The provided input member functions are in fact used to determine how much the received input resembles a press or a hold command from the user. In order to determine the received action from the user, the falling or rising edge of the input signals is used to set the value of a counter. This counter is in fact the input of the FLS. All the decisions made by the controller are the output of the FLS. A vital step in designing the controller is to gather data from the user and form a dataset which can be later used in fine-tuning of the parameters. The aforementioned fine-tuning is possible by calculating a gradient vector which provides a measurement of how well the system is modelling the input/output data for a set of parameters. Afterwards the calculated gradient vector is used to update the parameters of the FLS using backpropagation. One of the most challenges faced during the training stage of the FLS is overfitting the available data, therefore it is logical to say that the number of iterations in the training process should be limited. Considering the fact that the gathered dataset is user dependent, and also the fact that gathering an excessive number of data points from the user can lead to the discontent of the user, the initial values of the parameters of the FLS should be set in such a way that even by having few data from the user and few possible iterations, the values of the parameters will converge to that of the desired values. Finally, it is important to save some data points in the testing stage to guarantee the acceptable performance of the controller.

##### Acknowledgment

The authors would like to thank University of Isfahan Center of Entertainment Industry [[1]](#footnote-1)for their support in the development and implementation of the controller.

##### References

1. Calvert, S.L., Staiano, A.E. and Bond, B.J., 2013. Electronic gaming and the obesity crisis. *New directions for child and adolescent development*, *2013*(139), pp.51-57.
2. *https://en.wikipedia.org/wiki/Exergaming.*
3. Peng, W., Lin, J.H. and Crouse, J. Is playing exergames really exercising? A meta-analysis of energy expenditure in active video games. *Cyberpsychology, Behavior, and Social Networking*, *14*(11), 2011, pp.681-688.
4. KoenigHarold, G. Impact of game-based health promotion programs on body mass index in overweight/obese children and adolescents: a systematic review and meta-analysis of randomized controlled trials. *Childhood Obesity*, 2018.
5. Parry, I., Carbullido, C., Kawada, J., Bagley, A., Sen, S., Greenhalgh, D. and Palmieri, T., 2014. Keeping up with video game technology: Objective analysis of Xbox Kinect™ and PlayStation 3 Move™ for use in burn rehabilitation. *Burns*, *40*(5), pp.852-859.
6. *https://en.wikipedia.org/wiki/Game\_controller*.
7. *https://en.wikipedia.org/wiki/Dance\_Dance\_Revolution*.
8. *https://en.wikipedia.org/wiki/Kinect*.
9. Press, M.S.. PrimeSense Supplies 3-D-Sensing Technology to Project Natal for Xbox 360. *Press Release, 2010, March*.
10. Benzing, V. and Schmidt, M. Exergaming for children and adolescents: strengths, weaknesses, opportunities and threats. *Journal of clinical medicine*, 2018, *7*(11), p.422.
11. Hellendoorn, H., 1992. The generalized modus ponens considered as a fuzzy relation. *Fuzzy sets and systems*, *46*(1), pp.29-48.
12. Sugeno, M., Asai, K. and Terano, T., 1992. Fuzzy systems theory and its applications. *Tokyo Institute of Technology*.
13. Lixin, W. Course in Fuzzy System and Control, 1997.
14. Ram, M. ed., 2018. *Advanced Fuzzy Logic Approaches in Engineering Science*. IGI Global.
15. *https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html.*
16. *https://www.mathworks.com/help/fuzzy/neuro-adaptive-learning-and-anfis.html.*

1. http://uicvgames.ir [↑](#footnote-ref-1)