An Artificial Neural Network (ANN) in Electronic Product Design

Tom Page

Loughborough Design School, Loughborough University
Epinal Way, Loughborough, United Kingdom, LE11 3TU
T.Page@lboro.ac.uk

Abstract: This paper reports on the development of a wearable gesture recognition device for communicating with children that have acute communication difficulties. The device was designed with the aid of an artificial neural network (ANN) to develop and refine the physical gestures made by the user. The findings of the evaluation of the data handling are discussed with a view to improving on both its functions in gesture recognition and further development of its database.

Keywords: Artificial Neural Network, Product Design, Gesture Recognition.

1. Introduction

An ‘artificial neural network’ is, as it suggests, is a manmade system designed to mimic the cognitive ability of an animal brain. Gurney (1997) provided a neural model of the way in which artificial neural networks (ANNs) operate, it is estimated that the human brain comprises around 100 billion neurons. These neurons combine to make highly complex networks, as each neuron may have up to 10,000 connections. They communicate with each other via electrical impulses thus making up the brain (Drachman, 2005). Interneuron connections are mediated by synapses which are located on the end of dendrite cells that feed into the cell body, the soma. Here the synaptic inputs are integrated or summed together in some way and if the resultant is greater than a threshold level then the neuron will ‘fire’ a voltage impulse (Gurney, 1997). This output then feeds into other neuronal inputs.

Unlike conventional alphanumeric, equation-based computing, artificial neural networks provide for adaptability and ‘learning’ within a computer program. However, where biological and artificial neurons differ is in their ability to make each input more or less important than others through use of a weighting technique. Depending on the synaptic connection, each synapse’s electrochemical response can be polarised so that it is either excitatory (designed to promote synaptic firing) or inhibitory (designed to discourage synaptic firing). Additionally these ‘weights’ can change over time and this change in weight is interpreted as learning (Russell and Norvig, 2010).

In artificial neural networks, the equivalent to the neuron is called as a node or unit. A simple node works on the same level of understanding as biological neurons just given. As artificial neurons are currently based on digital inputs, the weighting technique differs slightly. Inputs, as in conventional computing, are still essentially made up of 1’s and 0’s. According to Gurney (1997), to achieve ‘weighting’, input data is passed through individual equations:

Input + Weight = Weighted Input

e.g. 1 (input) + 5 (weight) = 6 (weighted input)

These weighted inputs are then summed together and passed through a threshold limit. If the resultant is higher than the threshold then the node will produce a 1 if not then it will produce a 0. This type of artificial neuron is the simplest and historically earliest. It was put forward by McCulloch and Pitts (1943) and is known as a *threshold logic unit.*

In the brain, learning is achieved by neurons becoming more or less sensitive to certain synaptic responses. This can be emulated in artificial neurons by altering an input’s weight. The modification of weights is usually done automatically through use of comparison of data to classification code (Gurney, 1997). For example, if one considers pattern recognition of text, say a handwritten paper was scanned and inputted into the neural network. The ANN is told to analyse the individual characters looking for specific characteristics e.g. roundness, how many straight lines there are, are there any holes etc. This information becomes the input data for the neural network. The network is then given the output data (the correct translation of the inputted data). Through use of *back propagation* the neural network will attempt to work out how to get from the output to the input and adjust the weights accordingly (Lampinen, Laaksonen, and Oja, 1998).

This process is called ‘training’, effectively allowing the neural network to ‘learn’. All artificial neural networks need to go through this process hundreds of times before they can start to make logical and accurate assumptions (Gurney, 1997). If you have ever used voice recognition software you will be aware that before you can start using it effectively that you must read from passages of text. Here you are literally teaching and familiarising the software to the way that you speak and pronounce words (Van Daalen, and Bots, 2010).

2. ANNs in Product Design

Since the early 1990s ANNs have been applied in engineering and product design (Gero, 1990).Chu, Shoureshi and Tenorio (1990) suggested an ANN approach for systems analysis and synthesis. Brown and Eisenhardt (1995) provided an extant review of research directions in product development and ANNs. In a more closely related way Fels and Hinton (1993) compared an ANN over a hidden Markov model used in interfacing a data glove and a speech synthesiser.

According to Bolle, Connel, Pankanti, Ratha and Senior, (2004), biometric identification, or *biometrics,* refers to identifying individuals based on their distinguishing characteristics. More precisely, biometrics is the science of identifying a person based on a physiological or behavioural characteristic. Biometrics is playing an ever increasing role in the modern world particularly in security devices. This is in part, down to the use of ANNs and their ability to be able to perform pattern recognition with qualitative data.

Retinal scanners are now commonplace in security facilities and have more recently been introduced into airports. They work by taking a photograph of the blood vessels in the retina. Just like a fingerprint, the pattern of blood vessels in the eye is unique and can be used to identify a person accurately. An ANN will scan the pattern on the retina and create its own algorithms to define and link the pattern with a user. As retina patterns are unique, this makes it a perfect tool for security. Voice recognition software is now widely available (Porta, 2001). Products such as Dragon® or Windows Vista® use ANNs to train a voice recognition program to a user’s voice by reading out loud pre-written passages of text.

Body scanners have become commonplace in airports and are used to detect foreign objects and possible contraband such as weapons, bombs and drugs. Body scanners work on a two-part system that uses x-rays and millimetre waves to quickly produce a non-invasive but penetrating scan of the human body. ANNs in the scanner are used for the analysis of the imagery produced to detect foreign or contraband items. Using pattern recognition tools, objects such as bottles, guns and other items can be located and highlighted. Further development in body scanners in airports is to integrate electronic noses(Gardner, and Yinon, 2004). This would further enhance the detection of liquid and plastic explosives as these substances produce very slight but distinguishable scents. It’s likely that ANNs will play a key role in separating and identifying smell.

Over the past five years computer companies have been introducing biometric hardware in their products. Products such as fingerprint locks now come as standard on some laptops. The past couple of years have seen the integration of facial recognition in digital cameras. When these cameras first came out they were designed to highlight and show any faces that the camera could detect. This function is a useful indicator to detect if everyone in the photograph is looking at the camera(Curran, Xuelong, and McCaughley, 2005). However further advances in the technology have allowed increased pattern recognition to also check that everyone is smiling too. ANNs have been used in the development of this technology and trained to understand the general features that make up a face and also its expressions (Thorsteinsson, 2012).

Digital media has seen some of the biggest advances in ANNs as usually the people who are writing the code are also the ones who are using it (Lin, Lai and Yeh, 2007). Many computer games have claimed to include artificial intelligence (AI) to control non-playable characters (NPCs). While these NPCs may appear to look like they act of their own accord, it is usually the case that their movement has been scripted and is based on a player activating a trigger to set off a predefined response (Schifferstein, Otten, Thoolen, and Hekkert, 2010). In the past couple of years however, ANNs have started being implemented in NPC design. This has led to a new generation of gaming where non-playable characters start to learn the behavioural patterns of the player making for more challenging and realistic game play (Bourg, and Seemann, 2004).

Medicine has seen huge benefits from ANNs. They are particularly suited to diagnosing very complex diseases or finding solutions that have no algorithmic solution. To this end drug development and patient diagnosis have improved considerably with the use of neural networks. Two major advances in medicine have been in the detection of coronary artery disease and in the processing of electroencephalogram (EEG) signals (Malmgren and Niklassan, 2000).

ANNs are currently not widely used in everyday products available on the market. This is in part due to the fact that they are quite complicated systems to write and set up. However, as more research and development is being done into the area we are now starting to see more products available to on the open market. Sudoku Grab is one of over 350,000 ‘Apps’ available for download on the Apple iPhone®. The App has received much critical acclaim and even appears on an advert for the iPhone showcasing some of the best Apps currently available. Sudoku Grab is an App that takes advantage of the iPhones camera to take a picture of an incomplete Sudoku grid and automatically fill in the blanks in real time. Chris Greening, the developer of the App currently runs a blog on Blogspot (Greening, 2009) which talks through how he created the program.

3. Gesture Recognition Device

The ANN was primarily developed to explore different methods of handling input data for the gesture recognition ‘Articulate’ developed at Loughborough Design School. Articulate was designed for children with severe difficulties in communicating verbally and do not have the acute motor abilities that are required for hand gestures found in British Sign Language (BSL) (Starner and Pentland, 1995). It provided for those that do not possess the cognitive abilities required to differentiate between often subtle BSL gestures. Articulate was designed with these children in mind, aiming to alleviate some of the stress and anxiety that verbal communication can cause (Page, 2011).

Oscar was one such child that has had a great influence on the product, he is an eleven year old boy who has acute Cerebral Palsy (CP). CP is an umbrella term used for a variety of motor disorders, often associated with either involuntary spastic or tight and stiff muscles (NHS, 2008; Peacock, 2000). Oscar has extremely limited motor abilities and as a result cannot walk or talk; additionally many of his muscles (predominately in his arms) are constantly clenched and held into his body. Many hours were spent with Oscar observing his communicative abilities and it was determined that with the appropriate stimuli and encouragement he would be able to learn a gesture as shown in and to associate it with a word or situation.

Through working with children like Oscar, Articulate was developed as an autonomous device to be worn on the back of the hand. The premise of the device is relatively simple; by using accelerometers (a component used to measure acceleration in x, y and z translation) inside of the hand device, movement can be recorded and sent to a processing unit which looks for a pre-programmed gesture, and if it finds one it provides an audiovisual (AV) response in both the hand device and speaker unit.

4. Data Acquisition

During product development, an alternative system was developed to simulate the hand device. Two methods were used in the electronic prototyping stage. The first made use of the Arduino microcontroller board. The second made use of a Nintendo Wii® Mote and LabVIEW® programming software.

Arduino is an open-source physical computing platform (Mellis, 2007) used by developers globally to create and experiment with new inventions. For the first Articulate prototype, the Arduino microcontroller board was fitted with an accelerometer. A program was written which read a stream of data from the accelerometer which was read as three sets of numbers which represented acceleration in the X, Y and Z planes with data ranging from 0-999 where 500 denotes no acceleration. The data provided by the accelerometer is very useful in relation to gesture recognition (see Table 1). However the data in numerical form is very hard to analyse and interpret.

Table 1. Acceleration data in the X, Y and Z planes

X	Y	Z
564	599	320
558	596	318
556	594	326
557	592	336
551	591	338
540	595	339
524	598	338
509	598	339
504	599	334
511	593	332
513	594	332

Here it is much easier to see a change in acceleration and note when a definite movement has been made. The data read from the accelerometer made it possible to work out the device's orientation when stationary and calculate velocity and distance through use of integration. The second method of reading data was to use a Nintendo Wii® Mote. The Wii® Mote is a wireless remote control device that makes use of accelerometers through a Bluetooth connection. The Wii® Mote can be easily connected to a computer via Bluetooth and used as an input/output device using LabVIEW® graphical programming software.

A program was written to interface with the Nintendo Wii® Mote and an ANN was implemented in LabVIEW®, a graphical programming environment. The main advantage of using was LabVIEW® that it greatly reduced the amount of time required to write a program in the C++ programming language. Thus greater experimentation and optimisation was done within the same timeframe. It was possible to plot a graph similar to the one made for the Arduino development board but with the main difference being that it was a continuous live feed (Page, Thorsteinsson and Niculescu, 2009).

5. Rough Gesture Recognition

Analogue input devices provide continuous data which can only be measured to within a certain level of accuracy (Nguyen, 1999*)*. The human body can be considered an analogue input system when taking measurements from it (Russell and Norvig, 2010). This poses a problem when recording someone repeating a hand gesture in that they will never perform the exact same gesture twice. A test user was asked to perform a simple hand movement three times whilst holding an accelerometer. The user was told to try and repeat the movement as accurately as possible each time.

The findings of this test show that although the results are similar each time, i.e. the velocity and size of the gesture were alike, when plotted as discrete data it is clear that the gestures were not the same each time. This rough duplication of gesture in pattern recognition is the main issue that needs to be tackled in the product's development. Articulate was needed to be designed such that it can identify a gesture that will not be replicated exactly the same every time.

6. Limitations

Initial experiments with the Arduino board and Wii® Mote highlighted that the input data from the accelerometers was unclean and noisy. As the device was left stationary the results should have produced a perfectly straight line, any deviation to this suggests that there is some interference affecting the accelerometer. This may have been due to electrical or magnetic interference but it is more likely that it was caused by quality of the components. As the final product had to keep within a limited budget however, more expensive components could not be brought in as this would raise the cost of the final product, therefore a workaround had to be designed. The most efficient and accurate method that was developed for this approach was to use a polynomial line of best fit over the data and use the data from that to establish gesture recognition.

A polynomial curve was used to clean the data. This method is very good at smoothing the data making it appear more continuous, however it may be noted that where the data has deviated too far away from its stationary figure of around 300 the line has become curved. Another example of this smoothing process can be seen in (bold curve).

In this example, the accelerometer was being moved backwards and forwards in a smooth motion. Considering the raw data (shown lightly) much noise can be seen in the data. By adding the polynomial curve a much more natural dataset is developed.

7. ANN in Handling the Data

There are numerous ways that pattern recognition can be achieved when trying to identify a hand gesture such as the use of Hidden Markov Models (Erol*et al*, (2007); Starner and Pentland, (1995); Rozendaal, and Keyson, (2006)). Articulate makes use of the ANN in handling the data for gesture recognition and required datasets for perfectly performed gestures. In order to achieve this, gestures were performed and repeated numerous times and a mean average was taken of the results and used as a 'perfect gesture' (see Figures 1 and 2). For the purposes of test, two perfect response gestures were created. The first was the BSL sign for "Hello" and the second for "Goodbye".

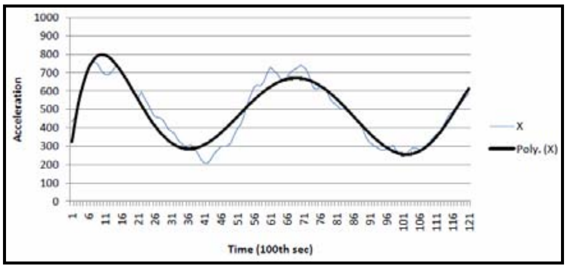

Figure 1. Graph showing 'cleaned' data

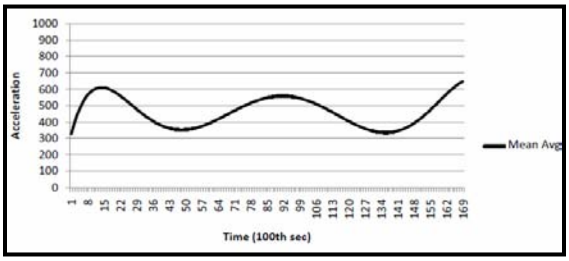

Figure 2. Graph showing perfect response for "Goodbye" in X plane

8. Graphical Comparison - Data

This method worked on the idea of comparing the raw input data to a 'percentage-like' value of each perfect response. If the percentage-like value was high enough, then the ANN would provide an output related to that gesture.

For the purpose of reporting the work done in developing Articulate only movement data from the X plane was used to prove whether the system would work. For full gesture recognition the same process was undertaken in the Y and Z planes too. The system works by first taking the raw input data and plotting it on a graph measuring acceleration over time. The graph is then given a polynomial line of best fit to help clean the noisy data.

The same is also done with the perfect responses for "Hello" and "Goodbye". After this stage, the input data is compared to both responses for "Hello" and "Goodbye" by using a number match operation. This operation works by transposing the input data over the perfect response data and for each increment of time counting how many times the data is the same as the line for either "Hello" or "Goodbye". There are very few points where the input signal crosses that of either response. When the number of matching points is turned into a percentage over time, very small 'percentage-like' figures are produced. Therefore a room for error buffer was needed to be created for each perfect response. This buffer creates a controlled area above and below each response as opposed to having a single line to perform the number matching. If the input line now falls inside this buffer it is counted as a number match.

The red line in this graph shows the original data for a perfect response. The grey area between the yellow and green lines is now the area that the program will consider to be a number match for that specific gesture. In this example, there are many more number matches as the program is no longer looking for an exact match or reproduction of a gesture which in turn yields a greater 'percentage-like'. The last part of this program is designed to make a decision on what gesture has been attempted in the input data, if any at all. This was achieved through use of a series of four logic statements that were set up as follows:

- If input > 70% like response "Hello" then output = Hello.
- If input > 70% like response "Goodbye" then output = Goodbye.

– If input > 70% like response “Hello” and “Goodbye” then output = Highest percentage-like.

– If input < 70% like response “Hello” and “Goodbye” then output = Nothing.

Logic statements set up in this fashion meant that a gesture had to be at least 70% like a perfect response gesture in order to be recognised. This ANN was tested against many different hand movements and gestures and managed to correctly identify most gestures that it was given. However, it struggled when a gesture was done at a different speed. In these instances the ANN generally gave an output of ‘Nothing’ as it could not discern one gesture from another.

9. Change in Direction

As discussed earlier, the main limitation with performing a graphical comparison is that gestures need to be repeated with some degree of precision and at a controlled speed to recognisable by the ANN. For most people, this would not be an issue but given that the device is designed primarily for children with severe motor disorders a more flexible solution had to be designed. The method here required processing the data received from the accelerometer first before performing any pattern recognition. This was achieved by working out the accelerometers direction of motion rather than its velocity.

By using direction of motion, time and distance could be eliminated from the gesture recognition process which meant that gestures could be performed much more loosely and as quickly or slowly as the user wanted. The same principle was applied to the design of Articulate, the only difference being that gestures were made in 3D rather than 2D which meant instead of having 8 possible directions of travel there were 26.

Direction of movement is relatively easy to work out from data provided by an accelerometer. By performing integration of the acceleration data then the velocity could be determined. Velocity is a measurement of the rate an object changes its position and is defined by its speed and direction (*Lehrman, 1998*). Knowing this, it is possible to work out directional data and convert it into a numerical response between 0-25. Gestures are then built up as a sequence of numbers between 0-25.

It was important that the product worked autonomously as the users, for the most part, would be unable to activate the product before each gesture. Therefore the ANN had to be constantly analysing the data as it came in and attempt to spot a gesture when it was performed. To stop the ANN constantly sending change of direction values for very slight movements, a limit gauge had to be used to ignore small changes in data. This limit gauge was designed so that unless the accelerometer was moved decidedly in one direction then it would not produce a value.

The ANN could much more easily distinguish the difference between attempted gestures. Moreover, gestures could be performed as fast or slow, vicious or soft, as the user wanted and the prototype would still be able to work out the gesture provided that the movement was correct. The main issue that arose from this was that if there was any deviation from the set path of a gesture then the system would not recognise the movement as a gesture.

10. ANN: Gesture Library Design

The development of an effective gesture library involved taking many different iterations of the same gesture and creating a ‘mean average’ of all the results as seen in the graphical comparison method. Another approach involved scripting a predefined path that the gesture must follow rigidly as seen in the change in direction method.

For the gesture library, an ANN was set up to ‘learn’ the gesture dataset. This learning process was achieved by setting up a list of gestures for the network to learn, for example: “Hello”; “Goodbye”; “Please”; and “Thank you”. The network was then programmed to look for certain characteristics in the data it received. Here the data passed through the ANN was analysed for characteristics such as magnitude (strength and speed of the gesture), direction, orientation and distance travelled. With this set up a very accurate picture of the gesture was computed as the data passed through the ANN was not directly read from the accelerometer. It was the processed change in direction data, so it received streams of numbers ranging from 0-25. This data was analysed to look for a rough recurring pattern in the number sequences. Once a set of rules was established for both methods it was then possible to start training Articulate. To do this, each of the desired gestures to be contained in the library was needed to be performed many times.

Each time a gesture was performed the ANN was then told which gesture had been attempted. Then from the list of rules set, the ANN attempted to work out how to constrain the gesture to each rule. The gestures were performed many times before the ANN was able to make any accurate assumptions as the data that is input is qualitative and will be different with each performance. Therefore each time a gesture was repeated the ANN altered and tweaked the constraints which defined it. The more times the Articulate was 'taught' then the better it became at identifying that gesture.

11. Conclusion

The *Articulate* product benefitted greatly from the use of the ANN and the author has considerable experience in developing and applying ANNs. They provide cognitive processing unlike that of algorithmic based computing. Their construction is based on that of a biological neural network and compute data in much the same way through use of a weighting ANN. Their main advantage is in their ability to handle large amounts of complex and convoluted data in an attempt to identify patterns or highlight information that would possibly be missed or ignored by conventional computing. A key element of this is the concept of fuzzy data and finding solutions of how to handle it.

Artificial neural networks proffer an innovative approach to functional design of products which would not simulate but embody intelligence in a product. These smart or intelligent products would help to improve quality of life in all areas. It is therefore in developing awareness of neural networks and their design applications, which will see them being used and developed further in product design.

Acknowledgements

The author acknowledges Tom Malloy for his hard work and dedicated contribution to the design and development of *Articulate*.

REFERENCES

1. BOLLE, R. M., J. H. CONNEL, S. PANKANTI, N. K. RATHA, A. W. SENIOR, **Basics of Bio metrics**, in Guide to Biometrics, Springer, New York, 2004.
2. BROWN, S. L., K. M. EISENHARDT, **Product Development: Past Research, Present Findings, and Future Di rections**, Academy of Management Review, Vol. 20, 1995, pp.343-378.
3. BOURG, D. M., G. SEEMANN, **Neural Networks**, in: AI for Game Developers, O'Reilly Media Inc., USA, 2004.
4. CHU, S. R., R. SHOURESHI, M. TENORIO, **Neural Networks for System Identification**, IEEE Control Systems. Vol. 10, 1990, pp. 31-35.
5. CURRAN, K., L. XUELONG, and N. McCAUGHLEY, **The Use of N eural Networks in Real-time Face Det ection**. Journal of Computer Sciences, Vol. 1, No. 1, 2005, pp. 47-62.
6. DRACHMAN, D. A., **Do We Have Brain to Spare?**, Journal of American Academy of Neurology, Vol. 64, No. 12, 2004-5. Available online: http://www.neurology.org/cgi/content/citation/64/12/2004
7. EROL, A., G. BEBIS, M. NICOLESCU, R. D. BOYLE, X. TWOMBLY, **Vision-based Hand Pose Estimation: A Review**, Computer Vision and Image Understanding Vol. 108, No. 1-2, 2007, pp. 52-73 Special Issue on Vision for Human-Computer Interaction.
8. FELS, S. S., G. HINTON, **GloveTalk: A Neural N etwork Int erface Be tween a DataGlove and a Speech Synthesi zer**, IEEE Transactions on Neural Networks. Vol. 4, 1993, pp. 2-8.
9. GARDNER, J. W., J. YINON, **Review of Conventional Electronic Noses and t heir Possible Ap plication to the Det ection of Explosives**, in Electronic Noses & Sensors for the Detection of Explosives. Kluwer Academic Publishers, Massachusetts, USA, 2004.
10. GERO, J. S. (Ed.), **Applications of AI in Engineering V**, Vol. 1: Design, Proc. 5th International Conference on Applications

of AI in Engineering, Computational Mechanics Publications & Springer-Verlag, Boston, MA, 1990.

11. GREENING, C., **Sudoku Grab - How does it all work?** 2009, Available at: http://sudokugrab.blogspot.com/2009/07/how-does-it-all-work.html, [Accessed 6 March 2011].
12. GURNEY, K., **Neural Networks – An Overview. An Introduction to Neural Networks**. UCL Press, London, 1997.
13. LAMPINEN, J., J. LAAKSONEN, E. OJA, **Pattern Recognition Problem**, in Image Processing and Pattern Recognition, edited by Cornelius T. Leondes. Neural Network Systems Techniques and Applications, Vol. 5, Academic Press Limited, London. 1998.
14. LEHRMAN, R. L., **Notions of Motion**, in Physics the Easy Way, 3rd ed. Barron’s Educational Series Inc., New York, USA, 1998.
15. LIN, Y.-C., H.-H. LAI, C.-H. YEH, **Consumer-oriented Product Form Design based on Fuzzy Logic: A Case Study of Mobile Phones**, International Journal of Industrial Ergonomics, Vol. 37 2007, pp. 531-543.
16. MALMGREN, B. A., L. NIKLASSAN, **ART Neural Networks for Medical Data Analysis and Fast Distributed Learning**, in Artificial Neural Networks in Medicine and Biology, Springer-Verlag, London, 2000.
17. McCULLOCH, W., W. PITTS, **A Logical Calculus of the Ideas Immanent in Nervous Activity**, Bulletin of Mathematical Biophysics, Vol. 7, 1943, pp. 115-133.
18. MELLIS, D. A., **What is Arduino?**, 2007, Available at: http://www.arduino.cc/en/Guide/Introduction [Accessed 13 March 2011].
19. National Health Service, **Symptoms of Cerebral Palsy**, UK, 2008, Available at: http://www.nhs.uk/Conditions/Cerebral-palsy/Pages/Symptoms.aspx [Accessed 19 October 2010]. What is Cerebral Palsy? In: Cerebral Palsy. Capstone Press, Minnesota, USA.
20. PAGE, T, **Prospects for the Design of Electronic Products in Second Life**, Studies in Informatics and Control, Vol. 20, No. 3, 2011, pp. 293-303.
21. PAGE, T, G. THORSTEINSSON, A. NICULESCU, **Management of Knowledge in a Problem Based Learning Environment**, Studies in Informatics and Control, Vol. 18, No. 3, 2009, pp. 225-262, ISSN: 1220-1766.
22. PORTA, M., **Vision-based User Interfaces: Methods and Applications**. International Journal of Human-Computer Studies, Vol. 57, No. 11, 2001, pp.27-73.
23. ROZENDAAL, M. C. D. V., KEYSON, **Designing User Interfaces with Gestures and Sound: Towards the Performance and Appeal of Voice Mail Browsing**. Journal of Design Research, Vol. 5, No. 1, 2006, pp. 96-115.
24. RUSSELL, S., P. NORVIG, **The Nature of Environments**, in Artificial Intelligence: A Modern Approach, 3rd ed. Pearson Education, Inc., New Jersey, United States, 2010.
25. SCHIFFERSTEIN, H. N. J., J. J. OTTEN, F. THOOLEN, P. HEKKERT, **The Experimental Assessment of Sensory Dominance in a Product Development Context**, Journal of Design Research, Vol. 8, No. 2, 2010, pp. 119-144.
26. STARNER, T., A. PENTLAND, **Visual Recognition of American Sign Language using Hidden Markov Models**. Proceedings of the International Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland, June, 1995.
27. THORSTEINSSON, G, **Piloting a Use of Graphic Tablets to Support Students Drawing within a Secondary School in Iceland**, Studies in Informatics and Control, Vol. 21, No. 2, 2012, pp.201-209.
28. Van DAALEN, C. E., P. W. G. BOTS, (2010) **Using a Systems Perspective to Design a Problem Solving Process**. Journal of Design Research, Vol. 8 No. 4, 2010, pp. 301-316.