A Neuropsychologically-Inspired Computational Approach to The Generalization of Cerebellar Learning

S. D. Teddy[1], E. M-K. Lai[2], and C. Quek[3]

Centre for Computational Intelligence, School of Computer Engineering,
Nanyang Technological University, Nanyang Avenue, Singapore 639798.
[1]sdt@pmail.ntu.edu.sg, {[2]asmklai,[3]ashcquek}@ntu.edu.sg

Abstract. The CMAC neural network is a well-established computational model of the human cerebellum. A major advantage is its localized generalization property which allows for efficient computations. However, there are also two major problems associated with this localized associative property. Firstly, it is difficult to fully-train a CMAC network as the training data has to fully cover the entire set of CMAC memory cells. Secondly, the untrained CMAC cells give rise to undesirable network output when presented with inputs that the network has not previously been trained for. To the best of the authors' knowledge, these issues have not been sufficiently addressed. In this paper, we propose a neuropsychologically-inspired computational approach to alleviate the above-mentioned problems. Drawing inspirations from the psychological aspects of the generalization of motor skill learning, the proposed "patching" algorithm strive to construct a plausible memory surface for the untrained cells in the CMAC network. We demonstrate through the modeling of human glucose metabolic process that "patching" of untrained CMAC cells offers a satisfactory solution to incomplete training data.

1 Introduction

The human cerebellum is a brain region in which the neuronal connectivity is sufficiently regular to facilitate a substantially comprehensive understanding of its functional properties. It constitutes a part of the human brain that is important for motor control and a number of cognitive functions [1], including motor learning and memory. The human cerebellum is postulated to function as a movement calibrator [2], which is involved in the detection of movement error and the subsequent coordination of the appropriate skeletal responses to reduce the error [3]. It has been established that the human cerebellum functions by performing associative mappings between the input sensory information and the cerebellar output required for the production of temporal-dependent precise behaviors [4]. The Marr-Albus-Ito model [5] describes how the climbing fibers of the cerebellum perform this function by transmitting moment-to-moment changes in sensory information for movement control.

The Cerebellar Model Articulation Controller (CMAC) [6] is a neural network inspired by the neurophysiological properties of the human cerebellum and is recognized for its localized generalization and rapid algorithmic computations. As a computational model of the human cerebellum, CMAC manifests as an associative memory network [7], which employs error correction signals to drive the network learning and memory formation. This allows for advantages such as simple computation, fast training, local generalization and ease of hardware implementation [2], and subsequently motivates the prevalent use of CMAC-based systems [8–10].

However, there are two significant issues associated with the effective utilization of the CMAC network. Firstly, it is difficult to fully-train the entire CMAC network. As CMAC is a local-learning network [11], comprehensive planning is required to generate a training data that ensures that all the network cells are trained. Furthermore, the construction of such a training data is not always feasible in cases such as modeling of ill-defined problems for which only limited amount of observations are available. Secondly, the behavior of a CMAC network is undefined in the untrained regions of the network. Although the learning convergence property of the CMAC network has been well-established, this merely implies that the stability of a CMAC-based system is guaranteed only within the well-trained regions of the corresponding CMAC network. Therefore, the stability of a CMAC-based system remains very much dependent on the careful planning of the network training. Furthermore, to the best of the authors' knowledge, there has been no previous attempt to resolve the issues of insufficient training data in the CMAC network.

Research into the neurophysiology of the human brain has established that the human cerebellum plays a significant role in the learning and acquisition of motor skills [12]. Scientific studies on skill learning have provided evidences that humans as well as animals have the innate ability to adapt and generalize skills acquired in a well-trained motor task to novel but similar situations [12–14]. There are generally two types of motor skill generalizations: *motor adaptation* [14] and *contextual interference* [12]. Motor adaptation refers to the capacity to adapt the execution of a well-trained motor task to changes in the external environment in which the task is to be performed. Contextual interference, on the other hand, refers to the ability of the training acquired on a specific motor task to influence the learning process of another novel but similar task. A number of physiological as well as psychological evidences supporting the notion of generalized learning in motor skill acquisition have been presented in the literature [15, 16]. This generalization capability offers an insight to human behavioral responses towards novel stimuli and changing working environments.

Drawing inspirations from the neurophysiology of the human cerebellum as a movement coordinator, as well as the related psychological aspects of generalization and adaptation in human motor skill acquisition, we propose a computational approach to alleviate the problem of insufficient training data in the CMAC network. This approach, referred to as ”patching” in the paper, constructs a plausible memory surface for the untrained memory cells in a CMAC network. The proposed ”patching” technique is subsequently evaluated on the modeling of human glucose metabolic process. This application is suitable particularly due to the fact that it is very difficult, if not impossible, to construct a dataset that is able to capture every combination of factors influencing the blood glucose level. The rest of the paper is organized as follows. Section 2 briefly describes the neurophysiological aspects of cerebellar learning and outlines the basic principles of the CMAC neural network. Section 3 presents the proposed patching technique. The modeling of human glucose metabolic process is presented in Section 4 to evaluate the effectiveness of the ”pacthing” technique. Section 5 concludes this paper.

2 CMAC Network and Cerebellar-based Learning Mechanism

The human cerebellum functions primarily as a movement regulator; and although it is not essential for motor control, it is crucial for precise, rapid and smooth coordinations of movements [2]. In order to effectively accomplish its motor regulatory functions, the cerebellum is provided with an extensive repertoire of information about the objectives (intentions), actions (motor commands) and outcomes (feedback signals) associated with a physical movement. The cerebellum evaluates the disparities between the formulated intention and the executed action and subsequently adjusts the operations of the motor centers to affect and regulate the ongoing movement. Studies in neuroscience has established that the cerebellum performs an associative mapping from the input sensory afferent and cerebral efferent signals to the cerebellar output, which is subsequently transmitted back to the cerebral cortex and spinal cord through the thalamus [17, 18]. This physiological process of constructing an associative pattern map constitutes the underlying neuronal mechanism of learning in the human cerebellum.

The human cerebellum has been classically modelled by the Cerebellar Model Articulation Controller (CMAC) [6, 7]. The model was proposed to explain the information-processing characteristics of its biological counterpart. The CMAC network functions as an associative memory that models the non-linear mapping between the mossy fiber inputs and the Purkinje cell outputs of the cerebellum. The massive mesh of granulle cell encoders in the cerebellum corresponds to an association layer that generates a sparse and extended representation of the mossy fiber inputs. The synaptic connections between the parallel fibers and the dendrites of the Purkinje cells formed an array of modifiable synaptic weights that motivates the grid-like CMAC computing structure. In the human cerebellum, these modifiable synaptic weights are linearly combined by the Purkinje cells to form the cerebellar output. In CMAC, the network output is computed by aggregating the memory contents of the active computing cells.

The CMAC network is essentially a multi-dimensional memory array, where an input acts as the address decoder to access the respective memory (computing) cells containing the adjustable weight parameters that constitute the corresponding output. In the CMAC network, the memory cells are uniformly quantized to cover the entire input space. The operation of the CMAC network is then characterized by the table lookup access of its memory cells. Each input vector to the CMAC network selects a set of active computing cells from which the output of the network is computed. Similarly, CMAC learns the correct output response to each input vector by modifying the contents of the selected memory locations.

This paper employs a generic cerebellar associative memory model which is based on a single-layered implementation of the CMAC network. Such an associative network has only one layer of network cells, but maintained the computational principles of the CMAC network by adopting a neighborhood-based activation of its computing cells. The layered cell activations in the original CMAC network contributed to three significant computational aspects: (1) smoothing of the computed output; (2) facilitating a distributed learning paradigm; and (3) activating the similar or highly correlated computing cells in the CMAC input space. These three modeling principles are similarly conserved in the single-layered cerebellar associative memory via the introduction of neighborhood-based computations. The activation of neighboring cells corre-

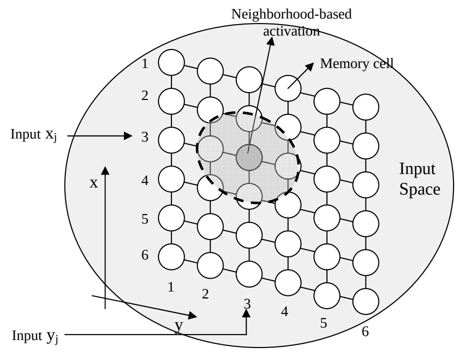

Fig. 1. The memory cells structure of a 2-input CMAC network

sponds to the simultaneous activation of the highly correlated cells in its multi-layered counterpart, and it also contributes to the smoothing of the computed output since the neighborhood-based activation process results in continuity of the output surface.

Figure 1 depicts the memory cell structure of such a single-layered implementation of a CMAC network. The single-layered CMAC network employs a *Weighted Gaussian Neighborhood Output* (WGNO) computational process, where a set of neighborhood-bounded computing cells is activated to derive an output response to the input stimulus. For each input stimulus $\mathbf{X}$, the computed output is derived as follows:

Step 1: Determine the region of activation
Each input stimulus $\mathbf{X}$ activates a neighborhood of CMAC computing cells. The neighborhood size is governed by the neighborhood constant parameter N, and the activated neighborhood is centered at the input stimulus.

Step 2: Compute the Gaussian weighting factors
Each activated cell has a varied degree of activation that is inversely proportional to its distance from the input stimulus. These degrees of activation functioned as weighting factors to the memory contents of the active cells.

Step 3: Retrieve the PSECMAC output
The output is the weighted sum of the memory contents of the active cells.

Following this, the single-layered CMAC network adopts a modified *Widrow-Hoff learning rule* [19] to implement a *Weighted Gaussian Neighborhood Update* (WGNU) learning process. The network update process is briefly described as follows:

Step 1: Computation of the network output
The output of the network corresponding to the input stimulus $\mathbf{X}$ is computed based on the WGNO process.

Step 2: Computation of learning error
The learning error is defined as the difference between the expected output and the current output of the network.

Step 3: Update of active cells
The learning error is subsequently distributed to all of the activated cells based on their respective weighting factors.

3 Generalization in The Neighborhood of Untrained CMAC Cells

It is not always feasible to generate a training data that ensures that all the memory cells in the CMAC network are trained. In such cases, the *empty cell's phenomenon* occurs whenever the test input falls within the clusters of untrained cells, resulting in an undesirable network output. However, this problem can be alleviated by constructing a plausible memory surface for the untrained cells of the untrained CMAC network cells. Such a construction process is referred to as "patching" in this paper.

The "patching" algorithm proposed in this paper is inspired by the neuropsychological aspects of human motor skill learning. In the research of human and animal motor skills, the *transfer of learning* [13] or *motor skill generalization* is a well-established property of skill acquisitions. It has been demonstrated that humans, as well as animals, have innate abilities to adapt and generalize skills acquired in a well-trained motor task to novel but similar situations [12, 13]. Such adaptation enhances the execution as well as shortens the learning curves of various motor skills.

Motor skill generalization capability can be broadly categorized into: (1) motor adaptation [14] and (2) contextual interference [12]. The motor adaptation process refers to the capacity to adapt the execution of a well-trained motor task to changes in the external environment where the task is to be performed [15]. Such generalization capability was demonstrated in a study conducted by Palmer and Meyer [20]. In that study, experienced pianists were first asked to learn a new piece of music and were subsequently asked to play a variation of the melody which required different combinations of hand and finger movements. The study eventually concluded that motor learning is not simply a matter of acquiring specific muscle movements, because experienced learners are able to transfer their skills to new situations that require them to produce the same general pattern of movements using different muscle groups [20]. Contextual interference, on the other hand, covers a broader scope of skill generalization. It refers to the ability of training acquired on a specific motor task to influence the learning process of another novel but similar task. Such generalization capability was demonstrated and studied in [21, 16]. Although much less is known about the exact neurophysiological processes underlying this motor generalization phenomena, psychological studies [12] have suggested that there is a correlation between the amount of skill transfer and the similarity in the natures of skill executions. Generally, as more similarities are observed between the two tasks, the greater is the influence of one over the other [13].

The local generalization characteristic of the CMAC network is based on the principle that similar inputs should produce similar outputs. Governed by this notion, this paper proposes a "patching" approach to the construction of a plausible memory surface for the untrained CMAC memory cells to alleviate the problem of insufficient training data in a CMAC network. The principle behind the proposed "patching" algorithm is the interpolation of memory surfaces from the trained memory cells to the regions of untrained memory cells. Starting from the outer edge of an untrained region, the memory content of an untrained memory cell is computed as the weighted average of the memory contents of its trained direct neighbors. For an arbitrary cell $c_{i,j}$ at the edge of an untrained region (see Figure 2), the "patched" value of the memory content $w_{i,j}$ is

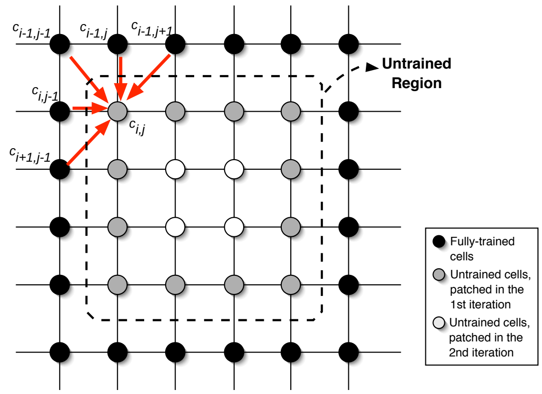

Fig. 2. The workings of the proposed ”patching” algorithm

computed as:

$$w_{i,j} = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} \frac{1/d_{k,l}}{\left(\sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} \frac{1}{d_{m,n}}\right)} w_{k,l} \tag{1}$$

where $d_{k,l}$ denotes the distance between the empty cell $c_{i,j}$ and its fully-trained neighboring cell $c_{k,l}$. The interpolated values are then propagated iteratively towards the center of the ”hole”. This is illustrated in Figure 2. This results in smooth transitions of the characteristic surface in the regions of clusters of untrained cells.

4 Case Study: The Modeling of The Human Glucose Metabolism

Diabetes is a chronic disease where the body is unable to properly and efficiently regulate the use and storage of glucose in the blood. This resulted in large perturbations of the plasma glucose level, leading to *hyperglycemia* (elevated glucose level) or *hypoglycemia* (depressed glucose level). Chronic hyperglycemia causes severe damage to the eyes, kidneys, nerves, heart and blood vessels of the patients while severe hypoglycemia can deprive the body of energy and causes one to lose consciousness and can eventually become life threatening. Currently, the treatment of diabetes is based on a two-pronged approach: strict dietary control and insulin medication.

The key component to a successful management of diabetes is essentially the ability to maintain a long-term near-*normoglycaemia* state of the patient. With respect to this notion, the therapeutic effect of discrete insulin injections is not ideal for the treatment of diabetes as the regulation of insulin is an open-looped process. Continuous insulin infusion through an insulin pump, on the other hand, is a more viable approach to a better management of the blood glucose level due to its controllable infusion rate [22]. Such insulin pumps are algorithmic-driven, with an avalanche of techniques proposed, investigated and reported in the literature over the years [23, 24]. All such proposed methods required some forms of modeling of the glucose metabolic process of the diabetic patient before a suitable control regime can be devised.

In this section, the performance of the proposed ”patching” algorithm is evaluated on the modeling of the dynamics of the human blood glucose cycle. This application is suitable particularly due to the fact that it is very difficult to collect a dataset that is able to capture every combination of factors influencing the blood glucose level.

4.1 Materials and Method

The first step into constructing a model of the human glucose metabolic process is to determine the patient profile to be modeled. Due to the lack of real-life patient data and the logistical difficulties and ethical issues involving the collection of such data, a well-known web-based simulator known as *GlucoSim* [25] from the Illinois Institute of Technology (IIT) is employed to simulate a person subject to generate the blood glucose data that is needed for the construction of the glucose metabolism model. The objective of the experiment is to apply the CMAC network, both with and without the ”patching” algoritm, to the modeling of the glucose metabolism of a healthy subject.

The simulated healthy person, Subject A, is a typical middle-aged Asian male. His body mass index (BMI) is 23.0, which is within the recommended range for Asian. Based on the person profile of Subject A, his recommended daily allowance (RDA) of carbohydrate intake from meals is obtained from the website of the Health Promotion Board of Singapore [26]. According to his sex, age, weight and lifestyle, the recommended daily carbohydrate intake for Subject A is approximately 346.9g per day.

Since the glucose metabolic process depends on its own current (and internal) states as well as the exogenous inputs in the form of food intakes, it is hypothesized that the blood glucose level at any given time is a non-linear function of prior food intakes and the historical traces of the insulin and blood glucose levels. To properly account for the effects of prior food ingestions to the blood glucose level, a historical window of six hours is adopted. A soft-windowing strategy is adopted to partition the six-hours historical window into three conceptual segments, namely: *Recent Window* (i.e. previous 1 hour), *Intermediate Past Window* (i.e. previous 1 to 3 hour) and *Long Ago Window* (i.e. previous 3 to 6 hour). Based on these windows, three normalized weighting functions are introduced to compute the carbohydrate content of the meal(s) taken within the recent, intermediate past or long ago periods. Thus, inclusive of the blood glucose and insulin levels, there are total of five inputs to the CMAC network.

Based on the formulated hypothesis and the preprocessed glucose data generated from GlucoSim, a total of 100 days of glucose metabolic data for Subject A are collected. The carbohydrate contents and the timings of the daily meals were varied from day-to-day during the data collection phase. This ensures that the networks are not being trained on a cyclical data set, but are employed to discover the inherent relationships between the food intakes and the glucose metabolic process of a healthy person. The collected data set is partitioned into 2 groups: 80 days of data as training data and the remaining 20 days is for testing.

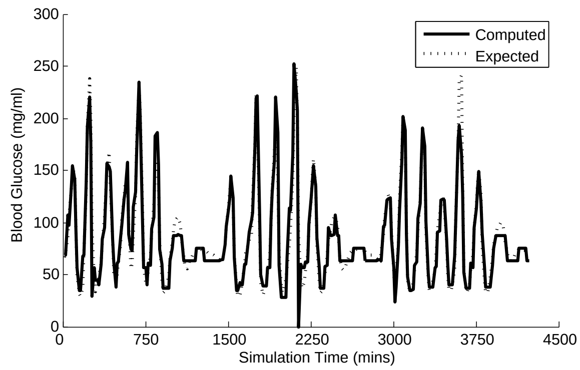

Fig. 3. Modeling results of the CMAC network on the glucose metabolic process of Subject A

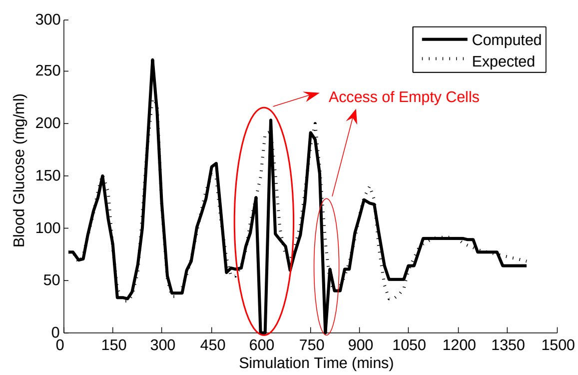

Fig. 4. Modeling results of the CMAC network on the glucose metabolic process of Subject A

4.2 Results

To model the blood glucose dynamics, a CMAC network with a memory size of 8 cells per dimension was constructed. The network was trained using the training dataset in 1000 training iterations, employing a learning constant of 0.1. At the end of the training iteration, a *Root Mean Square Error* (RMSE) of 6.3187 mg/ml and a *Pearson Correlation* of 98.97% were achieved. The trained network was subsequently tested to model the 20-days testing set. Figure 3 gives a 3-days snapshot of the modeling accuracy of the trained CMAC network.

The empty cells phenomena in the CMAC-based glucose metabolism model observed in this study are highlighted in Figure 4. Figure 4 depicts a one-day snapshot of the modeled blood glucose cycle during the testing phase. As the CMAC network was initialized to zero prior to the training process, the access of untrained CMAC cells result in zero network outputs. One can observe that empty cells phenomena results in poor and inaccurate performances of the CMAC glucose metabolic model.

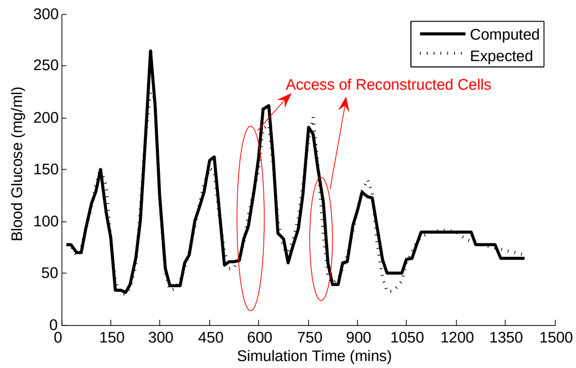

Fig. 5. Modeling results of the CMAC network on the glucose metabolic process of Subject A

Table 1. Testing results of the CMAC-based blood glucose modeling

CMAC Network	Maximum Error (mg/ml)	RMSE (mg/ml)	Pearson Correlation [%]
Before "patching"	190.006	10.7788	96.73
After "patching"	44.635	8.2548	98.08

The proposed "patching" technique is subsequently applied to the trained CMAC network to remove the untrained cells. Figure 5 depicts the performance of the "patched" network for the same day in the testing phase of Figure 4. It can be observed that the "patching" technique eliminates the empty cells phenomena and results in a significantly improved performance of the network. As a quantitative measure, Table 1 outlines the performances of the CMAC network before and after the "patching" process. Simulation results shown in both Figure 5 and Table 1 have justified the effectiveness of the "patching" technique in eliminating the problem of insufficient training data.

5 Conclusions

In this paper, we have presented a novel neuropsychologically-inspired computational approach to overcome the problem of insufficient training data in a CMAC-based system. An empty cells phenomenon occurs whenever the CMAC test inputs fall within the clusters of untrained CMAC memory cells, resulting in undesirable network output. The proposed "patching" technique alleviates this deficiency by interpolating the memory surfaces around the regions of untrained cells to construct a plausible memory surface for these untrained memory cells. The proposed technique was evaluated through the modeling of human glucose metabolic process. The experimentation results have sufficiently demonstrated the effectiveness of the "patching" technique, as significant improvements were noted in the performance of the "patched" networks. Further research in this direction includes a more detailed evaluation of the "patching" technique as well as the extension of the algorithm to more sophisticated problems.

References

1. Middleton, F.A., Strick, P.L.: The cerebellum: An overview. Trends in Cognitive Sciences **27**(9) (1998) 305–306
2. Albus, J.S.: Marr and Albus theories of the cerebellum two early models of associative memory. Proc. IEEE Compcon (1989)
3. Albus, J.S.: A theory of cerebellar function. Math. Biosci. **10**(1) (1971) 25–61
4. Kandel, E.R., Schwartz, J.H., Jessell, T.M.: Principles of Neural Science. 4 edn. McGraw-Hill (2000)
5. Marr, D.: A theory of cerebellar cortex. J. Physiol. London **202** (1969) 437–470
6. Albus, J.S.: A new approach to manipulator control: The Cerebellar Model Articulation Controller (CMAC). J. Dyn. Syst. Meas. Control, Trans. ASME (1975) 220–227
7. Albus, J.S.: Data storage in Cerebellar Model Articullation Controller (CMAC). J. Dyn. Syst. Meas. Control, Trans. ASME (1975) 228–233
8. Yamamoto, T., Kaneda, M.: Intelligent controller using CMACs with self-organized structure and its application for a process system. IEICE Trans. Fundamentals **82**(5) (1999) 856–860
9. Wahab, A., Tan, E.C., Abut, H.: HCMAC amplitude spectral subtraction for noise cancellation. Intl. Conf. Neural Inform. Processing (2001)
10. Huang, K.L., Hsieh, S.C., Fu, H.C.: Cascade-CMAC neural network applications on the color scanner to printer calibration. Intl. Conf. Neural Networks **1** (1997) 10–15
11. Miller, W.T., Glanz, F.H., Kraft, L.G.: CMAC: An associative neural network alternative to backpropagation. Proc. IEEE **78**(10) (1990) 1561–1657
12. Tomporowski, P.D.: The Psychology of Skill: A life-Span Approach. Praeger (2003)
13. Mazur, J.E.: Learning and Behavior. Pearson/Prentice Hall (2006)
14. Scheidt, R.A., Dingwell, J.B., Mussa-Ivaldi, F.A.: Learning to move amid uncertainty. Journal of Neurophysiology **86** (2001) 971–985
15. Lam, T., Dietz, V.: Transfer of motor performance in an obstacle avoidance task to different walking conditions. Journal of Neurophysiology **92** (2004) 2010–2016
16. Chen, Y., *et al.*: The interaction of a new motor skill and an old one: H-reflex conditioning and locomotion in rats,. Journal of Neuroscience **25**(29) (2005) 6898–6906
17. Houk, J.C., Buckingham, J.T., Barto, A.G.: Models of the cerebellum and motor learning. Behavioral and Brain Sciences **19**(3) (1996) 368–383
18. Tyrrell, T., Willshaw, D.: Cerebellar cortex: Its simulation and the relevance of Marr's theory. Philosophical Transactions: Biological Sciences **336**(1277) (1992) 239–257
19. Widrow, B., Stearns, S.D.: Adaptive Signal Processing. Prentice-Hall (1985)
20. Palmer, C., Meyer, R.K.: Conceptual and motor learning in music performance. Psychological Science **11**(1) (2000) 63–68
21. Weigelt, C., *et al.*: Transfer of motor skill learning in association football. Ergonomics **43**(10) (2000) 1698–1707
22. Fletcher, L., *et al.*: Feasibility of an implanted, closed-loop, blood-glucose control device. Immunology **230** (2001)
23. Schetky, L.M., Jardine, P., Moussy, F.: A closed loop implantable artificial pancreas using thin film nitinol mems pumps. Proceedings of International Conference on Shape Memory and Superelastic Technologies (SMST-2003) (2003)
24. Sorensen, J.T.: A Physiologic Model of Glucose Metabolism in Man and its Use to Design and Assess Improved Insulin Therapies for Diabetes. PhD thesis, Departement of Chemical Engineering, MIT (1985)
25. Illinois Institute of Technology: GlucoSim: A web-based educational simulation package for glucose-insulin levels in the human body. (Online: http://216.47.139.198/glucosim/gsimul.html)
26. Health Promotion Board Singapore. (Online: http://www.hpb.gov.sg)