THE POLE BALANCING PROBLEM

A Benchmark Control Theory Problem

Technical Report 7-01

Jason Brownlee
jbrownlee@ict.swin.edu.au

PhD Candidate
Master of Information Technology, Swinburne University of Technology, 2004
Bachelor of Applied Science, Computing, Swinburne University of Technology, 2002

Centre for Intelligent Systems and Complex Processes
Faculty of Information and Communication Technologies
Swinburne University of Technology
Melbourne, Victoria, Australia

July 2005

Introduction

The pole-balancing problem is a pseudo-standard benchmark problem from the field of control theory and artificial neural networks for designing and testing controllers on complex and unstable nonlinear systems. The intention of this work is to provide a canonical definition and description of this well studied problem domain. Further, this work will provided sufficient detail to permit a novice to the problem to be able to implement the problem for a variety of techniques. It will be shown that the problem if flexible in that it can be represented and addressed in numerous ways from devising controller units for the problem online and offline, to finding coefficient constants for existing linear models.

Pole Balancing

Control theory or control systems engineering is a mathematical and engineering discipline that is primarily concerned with understanding and controlling the behaviour of dynamical systems over time [40]. A control system is a set of interrelated components that form a system configuration that provides a desired response.

The pole-balancing problem is known by a number of common names; *pole-cart problem*, *broom balancer*, the *inverted pendulum problem* and the *stick balancer problem*. There are many variations on the pole-balancing problem, and the lack of uniform adoption of a standard problem definition is an issue that hinders useful benchmark comparison between reported results. Some seminal works have made efforts to standardise the problem [44] or in themselves have become de facto standards for the problem [1]. A concise somewhat canonical definition of the problem follows.

Problem Definition

The pole-balancing problem requires the proposal of a close-loop feedback control system with the desired behaviour of balancing a pole (an inverted pendulum) that is connected to a motor driven cart by a ball-baring pivot. The movement of the cart is restricted to the horizontal axis by a track, and the pole is free to move about the horizontal axis of the pivot. The state of the system is defined by four real values; the angle of the pole θ, the angular velocity of the pole $\dot{\theta}$, the position of the cart relative to the centre of the track x and the velocity of the cart $\dot{x}$. The output of the control system is a forward or backward movement for the cart as a fixed force.

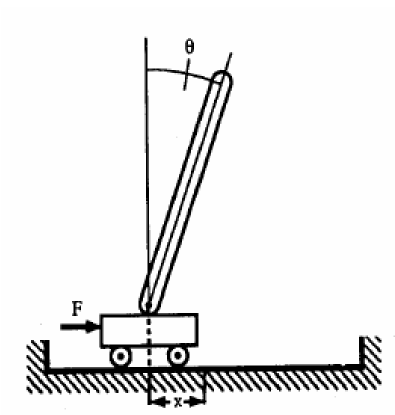

Figure 1 - One pole balancing problem, taken from [4]

System Parameters

This section describes the variables of the system, including both the state variables (inputs and outputs for a controller), as well as variables that govern the simulation of the system in a digital computer.

Symbol	Name	Description
θ	Pole Angle	The angle of the pole in radians
$\dot{\theta}$	Pole Velocity	The angular velocity of the pole in radians/second
$\ddot{\theta}$	Pole Acceleration	The acceleration of the pole in radians/second^2
x	Cart Position	Measured as a relative offset from the middle of the track (0.0) in meters
$\dot{x}$	Cart Velocity	The velocity of the cart in meters/second
$\ddot{x}$	Cart Acceleration	The acceleration of the cart in meters/second^2
g	Gravitational Acceleration	The acceleration due to gravity -9.81 meters/second^2 (where up is positive)
m_c	Mass of cart	1.0 kilograms
m_p	Mass of pole	0.1 kilograms
l	Pole length	The distance from the pivot to the poles centre of mass 0.5 meters (thus pole length is $2l$ = 1.0 meters)
t	Time	Measured in seconds
F	Force	The magnitude of force applied to the centre of the carts mass at time t. Typically a constant of ±10 newtons or ±1 newtons
h	Track Limit	±2.4 meters from the track centre (thus $2k$ = 4.8m)
r	Pole Failure Angle	±12° from 0° (approximately ±0.209 radians)
τ	Time Step	The discrete integration time step for the simulation, conventionally Δ = 0.02 seconds (50 Hertz)

System Constraints

1. The pole must remain upright within $\pm r$ the pole failure angle
2. The cart must remain within $\pm h$ of origin
3. The controller must always exert a non-zero force F

The simulation of the cart ends when either the pole exceeds the failure angle or the cart exceeds the limit of the track. The objective is to devise a controller that can keep the pole balanced for a defined length of simulation time. The controller must always output a force at full magnitude in either direction (bang-bang control). The system must be initialised with an initial position for the cart on the track and an initial angle for the pole. These initial values are typically a centred cart and balanced pole or are randomly selected from a Gaussian distribution with a mean of zero (position and angle) with a small standard deviation.

The problem definition is simplified with the following assumptions [41], which can be used when simulated in a digital computer:

1. The movement of cart and pivot is completely frictionless[1]
2. The electrical system for response is instantaneous
3. The wheels of cart do not slip
4. The motor torque limit is not encountered

Equations of Motion

The following provides the most common form of the equations for motion for both the pole angle and cart position.

$$\ddot{\theta}_t = \frac{g\sin\theta_t + \cos\theta_t\left[\dfrac{-F_t - m_p l\dot{\theta}_t^2\sin\theta_t}{m_c + m_p}\right]}{l\left[\dfrac{4}{3} - \dfrac{m_p\cos^2\theta_t}{m_c + m_p}\right]}$$

Equation 1 - Differential equation of motion of the pole

$$\ddot{x}_t = \frac{F_t + m_p l\left[\dot{\theta}_t^2\sin\theta_t - \ddot{\theta}_t\cos\theta_t\right]}{m_c + m_p}$$

Equation 2 – Differential equation of motion of the cart

The equations for motion are approximated numerically on a digital computer using Euler's method[2] with a time step τ of 0.02 seconds. The following provide the discrete time state equations for the pole and cart instantaneous positions and velocities.

$$x[t+1] = x[t] + \tau\dot{x}[t]$$
$$\dot{x}[t+1] = \dot{x}[t] + \tau\ddot{x}[t]$$

Equation 2 - Discrete time equations for the cart position and velocity

$$\theta[t+1] = \theta[t] + \tau\dot{\theta}[t]$$
$$\dot{\theta}[t+1] = \dot{\theta}[t] + \tau\ddot{\theta}[t]$$

Equation 3 - Discrete time equations for the pole angle and angular velocity

Linear solution

When the four inputs are provided as inputs for a controller, the system can be solved successfully with a linear artificial neural network with a single neuron called an ADALINE (Adaptive Linear Element) [7,8]. This solution can be expressed as a simple linear equation as follows.

$$F_t = F_m\,\mathrm{sgn}\left(k_1 x_t + k_2\dot{x}_t + k_3\theta_t + k_4\dot{\theta}_t\right)$$

Equation 4 - Linear equation for solving the one-pole problem with four inputs

---

[1] It should be noted that some work does include cart and pole friction coefficients, though work by Geva and Sitte [44] indicate that they are too small to have interesting effects or are cosmetic

[2] Given a differential equation, the derivative is replaced with a difference approximation, a time step size is selected and a series is calculated using a recursive scheme

Where F_m is a positive constant of the magnitude of the force F, *sgn* is a sign function[3], and (k_1, k_2, k_3, k_4) are system constant coefficients derived from bang-bang[4] control theory. The specific coefficients used depend on the masses of the cart and pole as well as the friction coefficients used. It has been noted that this linear control law will fail when the cart is far enough away from the centre of the track, further that coefficients can be found for the linear model easily using a random search [44].

Variations

The one pole-balancing problem given above is the most common form. Given that the one pole-balancing problem can be addressed with a simple linear equation, variations of the problem have been employed for which it is harder to model controllers. The following lists some of the more prevalent variations from the literature.

1. **Reduced Inputs One Pole Balancing**: The number of inputs can be reduced to just the current pole angle and cart position, requiring the controller to internally calculate the derivatives. The problem can be made more difficult by removing all inputs from the controller and using a consistent starting position (position 0 and vertical pole angle), the controller must then further estimate the cart position and pole angle.
2. **Two Poles Balancing**: Two poles of differing length are balanced on the cart. The required differing length causes the poles to react differently to the forces applied to the cart.
3. **Hinged Pole Balancing**: Also called the articulated pole, the joint pole problem and the pole on pole balance problem. The first pole is connected to the cart as described in the one pole problem; the second pole is attached to the top of the first pole by a second pivot. Like the two poles balancing problem, the poles must be of differing lengths.
4. **Changing Pole Lengths**: For all variations of the pole-balancing problem, it is possible to dynamically adjust the poles length over time (perhaps 1% per discrete time step). This obviously causes the mass of the pole to fluctuate with time adding additional non-linear complexities for the controller to address.

There are numerous other variations of the problem from adding friction coefficients, adding an additional dimension of freedom [32,34], and desiring the pole to balance toward a randomly moving particle [20].

Applications

The problem represents a useful prototypical laboratory idealisation of an unstable mechanical system [41]. It has been used as a simple case study in control theory for investigating controllers for unstable nonlinear problems as well as system recovery

---

[3] Sign Function: $\mathrm{sgn}(x) = \begin{cases} -1 : x < 0 \\ 0 : x = 0 \\ +1 : x > 0 \end{cases}$

[4] Bang-bang control refers to all or nothing control as opposed to proportional control, in this case full magnitude of force in either direction (±)

from bad initial conditions (in this case from a bad initial pole tilt angle). Specific scenarios can be created with one or two poles in specific locations that requires a complex series of force adjustments to stabilise. There are many linear and nonlinear solutions to this problem and its variants, meaning that there is no single correct solution, rather there are a set of controllers that work within defined constraints. This makes the problem interesting from the standpoint of testing techniques and tools in the digital computer for devising such controllers.

Perhaps there are three general approaches for a devising a controller for a dynamical system related to the amount of prior information available or accessible.

1. Prior information about the dynamics is available; a controller model can be devised and applied. An example of this is the linear model with known constant coefficients.
2. Prior information regarding the dynamics of the system is available though a complete model cannot be devised due to complexity or incomplete information. An example of this is devising a model, and letting a system tune the model.
3. There is little or no prior information and the dynamics of the system must be learned as a solution controller is being devised.

The dynamics of the pole-balancing problem and its variants is known a priori so that the problem can be simulated in a digital computer. The idea of using the problem as a test of an algorithm to devise a controller requires that specific prior knowledge be withheld. This is an important point, as the problem is sufficiently flexible to permit all three of the above approaches to be employed. The intent of this section is to highlight some of the seminal and more interesting controller applications to the pole-balancing problem and its variants.

Neural networks and reinforcement learning techniques have been the predominate application of this benchmark problem for control techniques [5,10,35,36,38,39,43]. A variation of the self organising neural network (SOM) was implemented by Hougen, Fischer, Johnam [22] in a real robot that addressed the one pole balancing problem in real time.

The one-pole problem can be addressed by conventional control theoretic techniques and has been exhaustively examined [6,41]. These approaches were also employed by Widrow and Smith [8] and Widrow [7] where control theoretic techniques were used to devise a linear control model for the one pole balancing system. A linear artificial neural network (ADALINE) was then trained to reproduce the controller. This work was extended by Guez and Selinsky [2,3] who used a two layer neural network to control the system. Their approach used a number of different training (teacher) schemes including a linear control law, a non-linear control law and a human teacher.

Miche and Chambers used an approach called BOXES [16] for controlling the cart. The systems state space was partitioned (pre-processed) into discrete regions called boxes to make the problem more manageable for the algorithm. Through experience with the dynamics of the problem, the algorithm updated force actions for each partition (box). Work by Barto, Sutton and Anderson [1] used an adaptive heuristic critic (AHC) to train a neural network to predict when the system will fail. The state space was discretized (quantised) like the BOXES approach into regions reducing the complexity of the problem when training the network.

This work on the BOXES algorithm was extended by Anderson [11] using real valued inputs on two-layer neural networks. A reinforcement learning approach was used that involved two networks, the first for estimating the system state which was trained using state variables at t-1 (the critic), and the second as an action network which incorporated the estimation output to make a decision as to the force to apply to the cart. This reinforcement learning approach called temporal difference (TD) was used to train the evaluation network and was further refined in later work by Anderson [12]. This was revisited by Dominic et al [42] where the neural network architecture was reused, though was trained using a genetic algorithm. The results showed that the algorithm prepared a controller in approximately the same number of trials, though it produced a controller that was somewhat (10%) more successful.

Fogel [18] applied a genetic algorithm to train a control system for the single pole balancer that operated in a to a similar manner to Barto et al's adaptive critic. The system used a number if lag terms in an evolved control that provided both a model to estimate the stimulus-response sequence, as well as control the system simultaneously in real-time.

An interesting approach to the pole balancing problem was taken by Tolat and Widrow [46] where a controller used input from a rough pixel image (5x11) of the systems current state. The approach used a supervised ADALINE neuron with the pixel inputs, which was successful at balancing a single pole. Troudet and Merrill [45] used a back-propagation neural network for the one pole problem where the system signals were made noisy or corrupted. The network controller was able to balance the pole and filter-out or manage the noise.

Evolutionary algorithm approaches have been combined with neural networks and reinforcement learning techniques and used to address variants of the pole balancing problem [9,15,17,20,23,24,26].

Wieland [4] successfully employed genetic algorithms to evolve the network weights of a recurrent neural network on the one pole, two pole, the hinged pole problem as well as varying length poles. The genetic algorithm used a fitness function to maximise the simulation time that an evolved network controller could balance the pole. The function included prior knowledge by using a penalty for the system the further the cart displaced from the centre of the track. Pasemann and Dieckmann [29] and Pasemann [27,28] also successfully employed a genetic algorithm to evolve recurrent neural networks to address the one pole balancing problem. Using their developed ENS technique both the structure and weights of recurrent networks were evolved for the problem in scenarios that used the position and angular derivatives and scenarios that did not.
Goldberg used an evolutionary technique called a learning classifier system (LCS) [19] on a variation of the one pole problem. The fitness function rewarded the controller for taking the cart from an initial starting position toward the centre of the track and bringing the cart to rest whilst keeping the pole balanced. Another interesting approach to controller design was by Koza [32], Koza and Keane [33,34]

where genetic programming (GP) was successfully applied to construct a controller for a two-dimensional and three-dimensional version of the pole balancing problem using LISP S-expressions (expression trees) .

Fuzzy logic controllers have been devised for the problem, prepared using both genetic algorithms [13,30,37] and neural fuzzy techniques [31].

Sammut implemented a visual version of BOXES as a java applet with source code [14]. Gomez and Miikkulainen provide digital simulations videos [25] of their enforced sub-populations (ESP) neural evolution system [24,26]. Pardoe, Ryoo, Miikkulainen also provide a video of a digital simulation [21] of their NEAT controller [20].

Final World

The problem represents an archetype credit-assignment problem when a controller must learn the dynamics of the problem. Given a sequence of force movements directed by a controller, the problem of determining which movements in the sequence caused a failure is a difficult and well-studied problem. As shown, a common approach to addressing this problem with neural network based controller is to use reinforcement-learning approaches to measure error between predicted state and historic measured state, such as the temporal difference method.

A common alternative approach shown to learning the dynamics through derivatives in real time was to use a genetic algorithm to prepare and test once-off network controllers that were improved through an evolutionary directed process of generate-and-test. This offline approach is typically less computationally efficient to the online training approach, though provides the same capabilities of learning the unknown complex dynamics and stabilising the system.

A good controller is not necessarily a controller that is capable of balancing the pole for a specified time until failure. A detailed review of the problem by Geva and Sitte suggest that other measures such as the time to stabilisation, the amount of oscillation and cart position from centre and pole angle over time may provide more insight into the quality of controllers [44].

The pole-balancing problem is a well studied somewhat standardised and interesting control theory problem that provides a useful benchmark problem for testing techniques capable of addressing complex linear and nonlinear dynamics of inherently unstable systems.

List of Equations

Equation 1 - Differential equation of motion of the pole 4
Equation 2 - Discrete time equations for the cart position and velocity 4
Equation 3 - Discrete time equations for the pole angle and angular velocity..............4
Equation 4 - Linear equation for solving the one-pole problem with four inputs 4

References

[1] A. G. Barto, R. S. Sutton, and C. W. Anderson, Neuron-like adaptive

elements that can solve difficult learning control problems *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 834-846, 1983.

[2] A Guez and J. Selinsky, "A neuromorphic controller with a human teacher," *Proceedings IEEE International Conference on Neural Networks,* pp. 595-602, 1988.

[3] A. Guez and J. Selinsky, A trainable neuromorphic controller *Journal of Robotic Systems*, vol. 5, pp. 363-388, 1988.

[4] Alexis P. Wieland, "Evolving neural network controllers for unstable systems," *Proceedings of the International Joint Conference on Neural Networks (IJCNN-91),* Seattle , pp. 667-673, 1991.

[5] B. E. Rosen, J. M Goodwin, and J. J Vidal, "State recurrence learning," *Proceedings of the first annual international network society meeting,* Boston, MA, USA , pp. 48, 1988.

[6] B. Friedland . *Control Systems Design - An Introduction to state-space methods*, New York, USA : McGraw-Hill Book Company, 1986.

[7] B. Widrow, "The original adaptive neural net broom-balancer," *Proceedings of the IEEE International Symposium Circuits & Systems,* pp. 351-357, 1987.

[8] B. Widrow and F. W. Smith, "Pattern-recognizing control systems," *Proceedings of Computer and Information Sciences (COINS)* , 1964.

[9] Borut Maricic, "Genetically Programmed Neural Network for Solving Pole-Balancing Problem," *Proceedings of the International Conference on Artificial Neural Networks (ICANN-91),* Helsinki-Espoo, Finland, pp. 1273-1276, 1991.

[10] C. C. Lee and H. R. Berenji, "An intelligent controller based on approximate reasoning and reinforcement learning," *Proceedings., IEEE International Symposium on Intelligent Control,* Albany, NY, USA, pp. 200-205, 1989.

[11] C. W. Anderson, "Strategy Learning with Multilayer Connectionist Representations," *Proceedings of the Fourth International Workshop on Machine Learning,* Irvine, CA, USA, pp. 103-114, 1987.

[12] C.W. Anderson, Learning to control an inverted pendulum using neural networks *IEEE Control Systems Magazine*, vol. 9, pp. 31-37, 1989.

[13] Charles L. Karr, "Design of an adaptive fuzzy logic controller using a genetic algorithm," *Proceedings of the fourth international conference on genetic algorithms,* San Diego, California, USA , 1991.

[14] Claude Sammut. Controlling Dynamic Physical Systems by Reinforcement Learning. [Online]

http://www.cse.unsw.edu.au/~claude/research/BOXES . 2002. 2005.

[15] D. Dasgulta and D. R. McGregor, " Evolving neurocontrollers for pole balancing," *Proceedings of the international conference on artificial neural networks (ICANN'93),* 1993.

[16] D. Miche and R. A. Chambers, "BOXES: An experiment in adaptive control," *Proceedings of the Second Annual Machine Intelligence Workshop,* University of Edinburgh, UK, pp. 137-152, 1968.

[17] D. R. McGregor, M. O. Odetayo, and D. Dasgupta, "Adaptive control of a dynamic system using genetic-based methods," *Proceedings of the 1992 IEEE International Symposium on Intelligent Control,* pp. 521-525, 1992.

[18] David B. Fogel. *Evolutionary Computation - Towards a new philosophy of machine intelligence*, New York, NY, USA: IEEE Press, 1995.

[19] David Edward Goldberg, Computer-Aided GAS Pipeline Operation Using Genetic Algorithms and Rule Learning 1983. University of Michigan.

[20] David Pardoe, Michael Ryoo, and Risto Miikkulainen, "Evolving neural network ensembles for control problems," *Proceedings of the 2005 conference on Genetic and evolutionary computation,* Washington DC, USA, pp. 1379-1384 , 2005.

[21] David Pardoe, Michael Ryoo, and Risto Miikkulainen. Evolving Neural Network Ensembles with NEAT. [Online] http://nn.cs.utexas.edu/keyword?NEATEnsembles . 2005. 2005.

[22] Dean F. Hougen, John Fischer, and Deva Johnam, "A neural network pole-balancer that learns and operates on a real robot in real time," *Proceedings of the MLC-COLT Workshop on Robot Learning,* pp. 73-80, 1994.

[23] Dirk Thierens and Leo Vercauteren, "A topology exploiting genetic algorithm to control dynamic systems," *Proceedings of the 1st Workshop on Parallel Problem Solving from Nature (PPSN),* pp. 104-108, 1991.

[24] Faustino Gomez and Risto Miikkulainen, "2-D pole balancing with recurrent evolutionary networks," *Proceedings of the International Conference on Artificial Neural Networks (ICANN-98),* Skovde, Sweden, pp. 425-430, 1998.

[25] Faustino Gomez and Risto Miikkulainen. Double pole balancing controller. [Online] http://nn.cs.utexas.edu/pages/research/espdemo/ . 2005. 2005.

[26] Faustino J. Gomez and Risto Miikkulainen, "Solving Non-Markovian Control Tasks with Neuro-Evolution," *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence,* pp. 1356-1361, 1999.

[27] Frank Pasemann, "Pole-Balancing with Different Evolved Neurocontrollers," *Proceedings of the 7th International Conference on Artificial Neural Networks,* pp. 823-829, 1997.

[28] Frank Pasemann, Evolving neurocontrollers for balancing an inverted pendulum *Network: Computation in Neural Systems*, vol. 9, pp. 495-511, 1998.

[29] Frank Pasemann and Ulf Dieckmann, "Evolved Neurocontrollers for Pole-Balancing," *Proceedings of the International Work-Conference on Artificial and Natural Neural Networks: Biological and Artificial Computation: From Neuroscience to Technology,* pp. 1279-1287 , 1997.

[30] G. D. Finn, Genetic Algorithm for Fine-Tuning Fuzzy Rules for the Cart-Pole Balancing System *Australian Computer Journal*, vol. 28, pp. 128-137, Nov, 1994.

[31] J-S Roger Jang, "Fuzzy controller design without domain experts," *IEEE International Conference on Fuzzy Systems,* pp. 289-296, 1992.

[32] John R. Koza, "Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems," Stanford University, Stanford, CA, USA, 1990.

[33] John R. Koza and Martin A. Keane, "Cart centering and broom balancing by genetically breeding populations of control strategy programs," *Proceedings of International Joint Conference on Neural Networks,* Washington, USA, pp. 198-201, 1990.

[34] John R. Koza and Martin A. Keane, "Genetic breeding of non-linear optimal control strategies for broom balancing," *Proceedings of the Ninth International Conference on Analysis and Optimization of Systems,* Antibes (France), 1990.

[35] M. E. Connel and P. E. Utgoff, Learning to control a dynamic physical system *Computer Intelligence*, vol. 3, pp. 330--3371987.

[36] Marcus Randall, Colin Thorne, and Clyde Wild, "A standard comparison of adaptive controllers to solve the cartpole problem," *Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems,* Brisbane, Qld., Australia, 1994.

[37] Michael A. Lee and Hideyuki Tagaki, "Integrating design stages of fuzzy systems using genetic algorithms," *Proceedings of the Second International IEEE Conference on Fuzzy Systems (Fuzz-IEEE'93),* San Francisco, CA USA, pp. 612-617, 1993.

[38] Michael I. Jordan and Robert A. Jacobs, "Learning to control an unstable system with forward modeling," *Advances in neural information processing systems 2,* pp. 324-331 , 1990.

[39] R. S. Bapi, B. D'Cruz, and G. Bugmann, Neuro-resistive grid approach to trainable controllers: a pole balancing example *Neural Computing and Applications*, vol. 5, pp. 33-44, 1997.

[40] Richard C. Dorf and Robert H. Bishop. *Modern Control Systems*, NJ, USA: Pearson Prentice Hall, 2005.

[41] J. Robert H. Cannon. *Dynamics of physical systems*, USA: McGraw-Hill Inc., 1967.

[42] S. Dominic, R. Das, D. Whitley, and C. Anderson, "Genetic reinforcement learning for neural networks," *International Joint Conference on Neural Networks,* Seattle; WA (USA), pp. 71-76, 1991.

[43] S. Geva and J. Sitte, "Performance of temporal differences and reinforcement learning in the cart-pole experiment," *Proceedings of 1993 International Joint Conference on Neural Networks (IJCNN '93),* Nagoya, Japan, pp. 2835-2838, 1993.

[44] Shlomo Geva and Joaquin Sitte, A cartpole experiment benchmark for trainable controllers *IEEE Control Systems Magazine*, vol. 13, pp. 40-51, 1993.

[45] T. Troudet and W. Merrill, Neuromorphic learning of continuous-valued mappings from noise-corrupted data *IEEE Transactions on Neural Networks*, vol. 2, pp. 294-301 , 1991.

[46] V.V. Tolat and B. Widrow, "An adaptive 'broom balancer' with visual inputs," *IEEE International Conference on Neural Networks,* pp. 641-647, 1988.