ORBIT CORRECTION STUDIES USING NEURAL NETWORKS

E. Meier*, Y.-R. E. Tan, G. S. LeBlanc, Australian Synchrotron, Clayton 3168, Australia

Abstract

This paper reports the use of neural networks for orbit correction at the Australian Synchrotron Storage Ring. The proposed system uses two neural networks in an actor-critic scheme to model a long term cost function and compute appropriate corrections. The system is entirely based on the history of the beam position and the actuators, i.e. the corrector magnets, in the storage ring. This makes the system auto-tuneable, which has the advantage of avoiding the measure of a response matrix. The controller will automatically maintain an updated BPM corrector response matrix. In future if coupled with some form of orbit response analysis, the system will have the potential to track drifts or changes to the lattice functions in "real time". As a generic and robust orbit correction program it can be used during commissioning and in slow orbit feedback. In this study, we present positive initial results of the simulations of the storage ring in Matlab.

INTRODUCTION

Control systems are crucial to ensure beam quality and stability. As accelerator systems become more complex, there is a real need for the development of more sophisticated control algorithms for beam tuning and beam-based control. In most cases, conventional controllers such as the PID algorithm can be used. Although they have the advantage of being conceptually very simple, they have several important limitations as the number of parameters to control increases [1]. These include necessary and time consuming re-measurements of the response matrix, re-tuning of the controller parameters whenever parameters outside the control loop are modified, and increased control imprecisions due to inaccurate measurements of the response matrix.

In order to overcome these issues, we investigate the application of neural networks to beam tuning. An example of a large scale control problem in accelerator physics is the orbit correction in the storage ring, where the response of the beam orbit is essentially a linear function of the corrector magnets. However, with a view to eventually utilize the system in other beam-based control applications, we investigate a system applicable to nonlinear systems. In the proposed scheme, an actor neural network learns to take appropriate control actions in order to minimize a long term cost function modeled by a critic neural network [2]. Such a system has the advantage of being auto-tunable, since it is trained online in a continuous fashion. This paper details the system, presents some simulation results and discusses system limitations and anticipated improvements.

* evelyne.meier@synchrotron.org.au

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks mimic the way biological neurons function. A neuron or node receives an input vector $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ from other nodes in the network. A weight is associated with each input component to reflect their relative importance, forming the weight vector $\mathbf{V}$. The weighted sum of the inputs $\sum_i v_i x_i$ then goes through an activation function ϕ, to give the neuron's response $\phi(\sum_i v_i x_i)$. Possible activation functions include hyperbolic tangent, Gaussian, sigmoid and linear functions [3].

Artificial neural networks are commonly composed of three distinctive layers; an input layer, a hidden layer and an output layer (see Fig.1), with n_{in}, n_{hid} and n_{out} neurons, respectively. Two weight matrices are associated with connections between the different layers; one for the connections between input and hidden layers, $V \in R^{n_{in} \times n_{hid}}$, and one for the connections from the hidden layer to the output layer, $W \in R^{n_{hid} \times n_{out}}$. The output of the network can then be written as [3]:

$$\mathbf{Y} = \mathbf{W}^T \phi(\mathbf{V}^T \mathbf{X}), \tag{1}$$

where X is the input vector to the network, and ϕ is the activation function of the nodes in the hidden layer. We note that the activation function of the output layer nodes corresponds to the identity function; the input neurons only serve to pass the information from the input layer to the hidden layer.

CONTROL APPROACH

Consider a general nonlinear process, written as [2]:

$$\mathbf{z_{k+\tau}} = f(\mathbf{\bar{z}_k}, \mathbf{\bar{u}_{k-1}}, \mathbf{u_k}, \mathbf{\bar{d}_{k+\tau-1}}), \tag{2}$$

where k is the current time step, τ is the system delay, $\mathbf{z_k}$, $\mathbf{u_k}$ and $\mathbf{d_k}$ represent the system output, input and disturbance vectors, respectively. We also define $\bar{\mathbf{z}}_k = [\mathbf{z}_k, ..., \mathbf{z}_{k-n+1}]^T$, $\bar{\mathbf{u}}_{k-1} = [\mathbf{u}_{k-1}, ..., \mathbf{u}_{k-n+1}]^T$ and $\bar{\mathbf{d}}_{k+\tau-1} = [\mathbf{d}_{k+\tau-1}, ..., \mathbf{d}_k]^T$. The system described by Eq. (2) is a function of the history of the n past intputs and outputs of the system.

Now let us define the following control objective or long term cost function [2]:

ISBN 978-3-95450-115-1

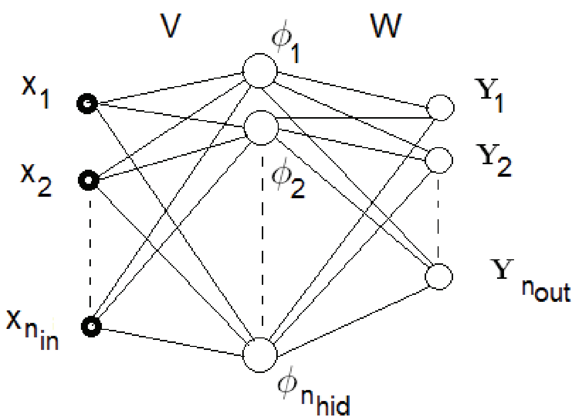

Figure 1: Schematic representation of an artificial neural network with three layers; input, hidden and output layers.

$$J_k = \sum_{i=0}^{\infty} \gamma^i r_{k+i}, \qquad (3)$$

where $\gamma > 0$ is the discount factor and r_{k+i} is the short term cost given by:

$$r_k = \mathbf{e}_k \mathbf{Q} \mathbf{e}_k^T + \mathbf{u}_k \mathbf{R} \mathbf{u}_k^T, \qquad (4)$$

in which the matrices $\mathbf{Q}$ and $\mathbf{R}$ are positive defined and the error term is given by:

$$\mathbf{e}_k = \mathbf{z}_k - \mathbf{z}_{k,ref}, \qquad (5)$$

where $\mathbf{z}_k$ is the actual system output and $\mathbf{z}_{k,ref}$ its reference signal. Thus, minimising J_k is equivalent to simultaneously minimising the long term error and correction signal $\mathbf{u}_k$.

To compute the optimal $\mathbf{u}_k$, we use the actor-critic scheme illustrated in Fig. 2. The critic neural network (CNN) computes $\hat{J}_k$, the estimate of the true J_k, based on the system output error $\mathbf{e}_k$; it receives the vector $\mathbf{X}_{cnn} = \mathbf{e}_k$ as its input (see Fig. 2). The actor neural network (ANN) then calculates the correction signal $\mathbf{u}_k$ based on the reference output signal $\mathbf{z}_{ref}$ and the input-output history of the system; it receives the vector $\mathbf{X}_{ann} = [\bar{\mathbf{z}}_\mathbf{k}, \bar{\mathbf{u}}_{k-1}, \mathbf{z}_{ref}]^T$ as its input.

Both neural networks are trained online, and it can be shown that the weight update law is, for the CNN [2] :

$$\mathbf{W}_{k+\tau}^{cnn} = \mathbf{W}_k^{cnn} - \alpha_{cnn} \gamma \phi_k^{cnn} (\gamma \hat{J}_k + r_k - \hat{J}_k), \qquad (6)$$

where α_{cnn} is the CNN learning rate and $\phi_k^{cnn} = \mathbf{W}_{cnn}^T (\mathbf{V}_{\mathbf{cnn,k}}^\mathbf{T} \mathbf{X}_\mathbf{cnn})$ is the corresponding output of the hidden layer nodes of the CNN. For the ANN, the weight update law is given by [2]:

$$\mathbf{W}_{k+\tau}^{ann} = \mathbf{W}_k^{ann} - \alpha_{ann} \phi_k^{ann} (\mathbf{e}_{k+\tau} + \hat{J}_k)^T, \qquad (7)$$

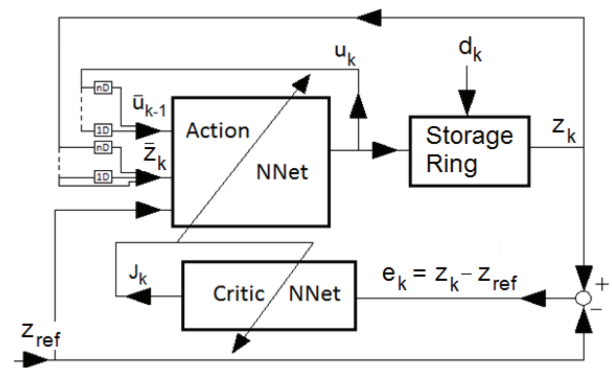

Figure 2: Actor - Critic control scheme. The ANN computes the appropriate correction based on the system's input and output history, while the CNN models the long term cost function.

where α_{ann} is the ANN learning rate and $\phi_k^{ann} = \phi^{ann}(\mathbf{V}_{ann}^T \mathbf{X}_{ann,k})$. Equation (7) shows that it is necessary to have a matching number of positions and actuators. Indeed, $W^{ann} \in R^{n_{hid} \times n_{out}}$, where n_{out} has for dimension the number of actuators; since $\phi^{ann} e_{k+\tau} \in R^{n_{hid} \times n_{var}}$, where n_{var} is the number of process variables, we must have $n_{var} = n_{out}$ for Eq. (7) to hold. Another assumption of this control scheme is that the response matrix is positive defined [2].

APPLICATION TO THE STORAGE RING

In what follows we show results obtained using Matlab to simulate the storage ring, which has a total of 42 horizontal correctors (3 for each of the 14 sectors) and 98 BPMs (7 per sector). We shall show the results obtained for a 3 BPMs - 3 correctors configuration for visual simplicity. BPM number in sectors 1, 3 and 7 were chosen along with and the last corrector in the same sectors.

The horizontal beam position having a linear response with the current in the corrector magnets, the positions at the chosen BPMs can be written as:

$$\mathbf{z}_k = \mathbf{z}_{k-1} + \mathbf{M} \mathbf{u}_k, \qquad (8)$$

where the horizontal position vector is noted $\mathbf{z}$ for consistency with previous notations, $\mathbf{M}$ is the response matrix, and $\mathbf{u}_k$ is the applied correction to the corrector magnets. According to Eq. (8), we can use $n = 1$ lag in Eq. (2).

Figure 3 below shows a typical simulation result, for which $Q = 0.01 I_3$ and $R = 0.05 I_3$, where I_3 is the 3×3 identity matrix. The discount factor was $\gamma = 0.5$, and the learning rates were $\alpha_{ann} = 0.05$ and $\alpha_{cnn} = 0.08$. All weight matrices were initialized to random numbers ranging from -0.4 to 0.4. Both the ANN and the CNN networks had 10 hidden nodes with hyperbolic tangent functions.

The first row of plots shows the actual and target horizontal position of the beam at the different BPMs in gray and black, respectively. The second row displays the cor-

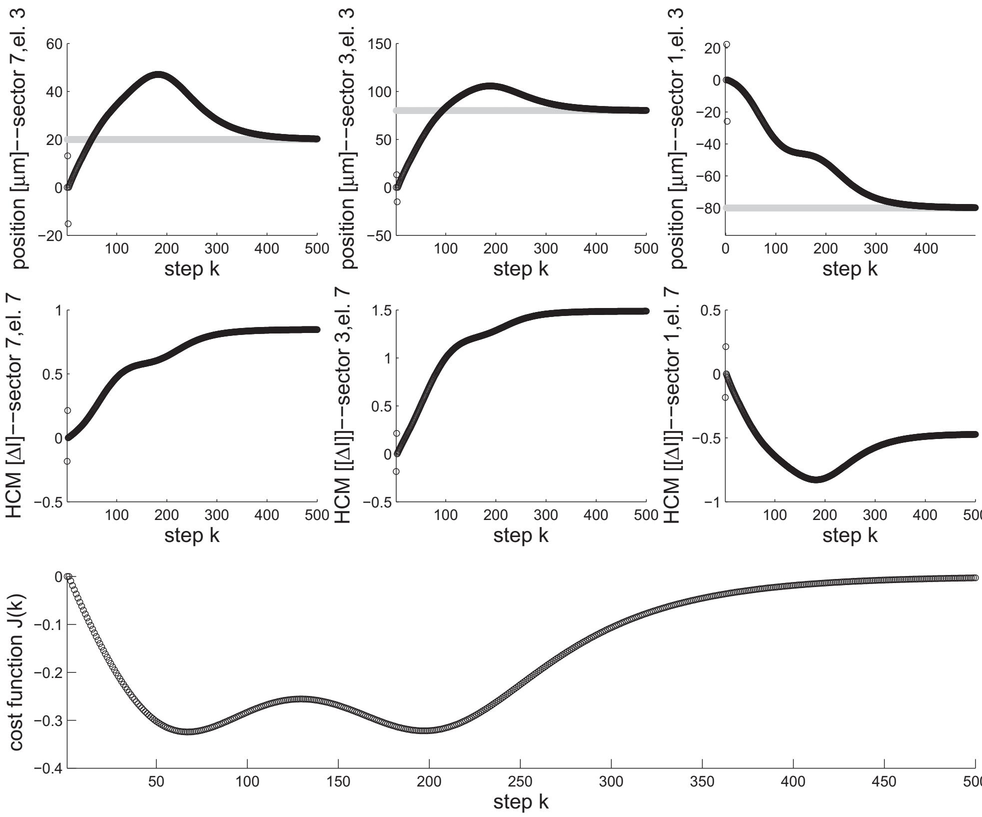

Figure 3: Example of a simulation run. Plots in the first row display the beam horizontal position, while plots in the second row give the corresponding correctors magnet response. The unique plot in row three displays the cost function $\hat{J}_k$ modeled by the CNN.

rector magnets' response. Finally, the plot in the third row displays the cost function $\hat{J}_k$ computed by the CNN. The initial horizontal positions of the beam were [0,0,0] μm and their set points [20,80,-80] μm.

The results show that the control system brings all positions towards their set points up to k ≅50, where the beam position in sector 1 crosses its set point. This results in an increase in the absolute value of the cost function $\hat{J}_k$, due to important changes in actuators' settings. The beam moves away from the target positions in sectors 1 and 3 until k ≅190, where the system has learnt to bring the beam its target position for all location. Desired settings are reached after 350 steps. We note that the system's response would improve for higher penalties on position errors rather than changes in actuators settings (i.e. higher values for the elements of $\mathbf{Q}$ rather than those in $\mathbf{R}$).

CONCLUSIONS AND FURTHER STUDIES

The system was successfully tested for the storage ring using Matlab. It has the advantage of avoiding a precise measurement of the response matrix in order to perform the correction. There are however significant problems such as the requirement that the response matrix be positive defined and that there are matching inputs and actuators. These limitations make it practically unusable in storage ring orbit control applications. Therefore efforts have been put into adapting the actor-critic scheme to the general multi-input multi-output case, using a modified expression of the cost function as described in [4].

REFERENCES

[1] J. Lygeros *Hierarchical, Hybrid Control of Large Scale Systems*, UCB-ITS-PRR-96-23, 1996.

[2] Q. Yang and S. Jagannathan. Control of nonaffine nonlinear discrete-time systems using reinforcement-learning-based linearly parametrized neural networks . *Transactions on systems, man and cybernetics - part B: cybernetics*, 39:994–1001, 2008.

[3] N. B. Karayiannis and A. N. Venetsanopoulos. *Artificial Neural Networks.* Kluwer Academic, 1993.

[4] J. Fu, He and X. Zhou. Adaptive Learning and control for MIMO systems based on adaptive dynamic programming. *IEEE transactions on neural netweorks*, 22:1133–1148, 2011.

ISBN 978-3-95450-115-1