Adaptive Stochastic Feedback Control of Resistive Wall Modes in Tokamaks

Zhipeng Sun, Amiya Sen and Richard Longman

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

11th Workshop on MHD Stability & Control
"Active MHD Control in ITER "
Princeton, New Jersey
Nov 6- Nov 8 2006

Outline

- Motivation
- Part I: System identification
- Part II: Adaptive output feedback control
- Part III: Neural network control
- Summary

Motivation

- Resistive Wall Mode (RWM) in tokamaks
  - Need for stabilization
- Past research on the suppression of RWM
  - Deterministic models used
  - No optimal control used
  - Fixed gain controllers without adaptive structure
  - PID control inadequate for multi unstable modes
- Sen, Nagashima and Longman’s work
  - Stochastic model used, optimal state feedback
- New methods as in the outline
  - State feedback control, output feedback control and its neural netowrk (NN) implementation

Part I

Online System Identification

Objectives

- A mathematical model should be estimated from experimental data
  - Should be able to estimate time-varying systems
  - Convergence time should be shorter compared to the inverse of the growth rate
  - Computational burden should be small

System models of a single unstable RWM

- State-space system model of a single unstable RWM

$$\dot{I}(t) = AI(t) + Bu(t) + DI_n(t)$$
$$\psi(t) = HI(t) + \psi_m(t)$$

For a RWM in DIII-D tokamak

$$A = \begin{pmatrix} -0.474 & -502 \\ -14.7 & 79.0 \end{pmatrix} \quad B = \begin{pmatrix} -531 \\ -16500 \end{pmatrix}$$

$$D = \begin{pmatrix} 100 \\ 30 \end{pmatrix} \qquad H^T = \begin{pmatrix} 1.40\times10^{-6} \\ 2.30\times10^{-6} \end{pmatrix}$$

- Difference equation model

$$A(q)\psi(k) = B(q)u(k) + C(q)e(k)$$

  - ➢Sampling rate is 1ms, *e(k)* is system noise
  - ➢*q* is the forward shift operator
- Noise modeling
  - ➢Total noise, approximately ½ to 1 Gauss, is evenly divided between the measurement noise and the plant noise
  - ➢RMS value of the plant noise is about 10^{-4} Weber
  - ➢RMS value of the measurement noise about 10^{-4} Weber

- Regression model setup

Input Sequence : $\{u(1)\ldots u(k)\ldots u(n)\}$, Output Sequence : $\{\psi(1)\ldots\psi(k)\ldots\psi(n)\}$

$\theta = \left(a_1\ a_2\ b_0\ b_1\ c_1\ c_2\right)$ $\varphi^T(k) = (-\psi(k)\ \ -\psi(k-1)\ \ u(k)\ \ u(k-1)\ \ e(k)\ \ e(k-1))$

use $\varepsilon(k) = \psi(k) - \varphi^T(k-1)\hat{\theta}(k-1)$ to approximate system noise $e(k)$

- Autoregressive system model

$$\psi(k) = \varphi^T(k-1)\theta$$

- Extended least square (ELS) method

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)(\psi(k) - \varphi^T(k)\hat{\theta}(k-1))$$

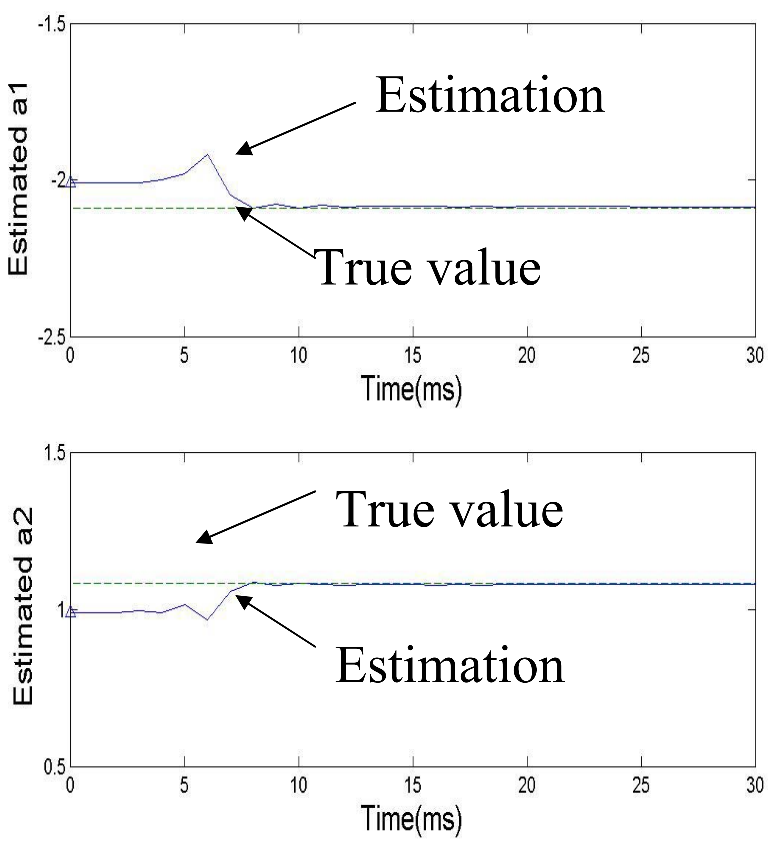
$$K(k) = P(k)\varphi(k) = P(k-1)\varphi(k)(I + \varphi^T(k)P(k-1)\varphi(k))^{-1}$$

$$P(k) = (I - K(k)\varphi^T(k))P(k-1)$$

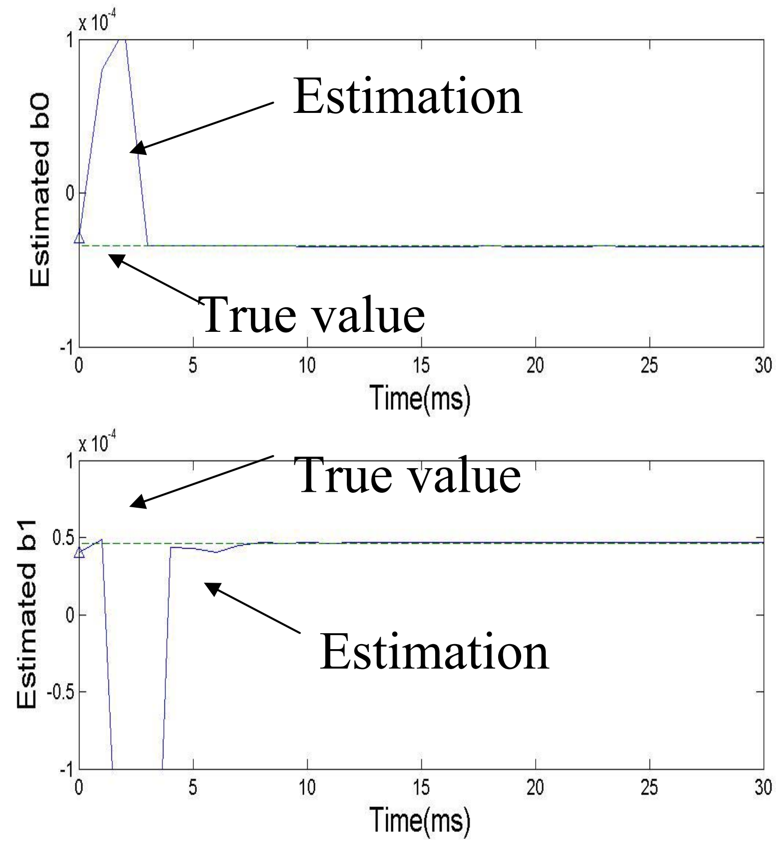
  - $\hat{\theta}$ is defined as the estimate of θ

Identification of the time-invariant model

Estimation of *A(q)*



Estimation of *B(q)*



ELS method for the time-varying system

- Real plasma systems are time-varying
  - Use a forgetting factor λ, $0 < \lambda \leq 1$, The ELS method becomes

$$\hat{\theta}(k) = \hat{\theta}(k-1) + K(k)(\psi(k) - \varphi^T(k)\hat{\theta}(k-1))$$

$$K(k) = P(k)\varphi(k) = P(k-1)\varphi(k)(\lambda I + \varphi^T(k)P(k-1)\varphi(k))^{-1}$$

$$P(k) = (I - K(k)\varphi^T(k))P(k-1)/\lambda$$

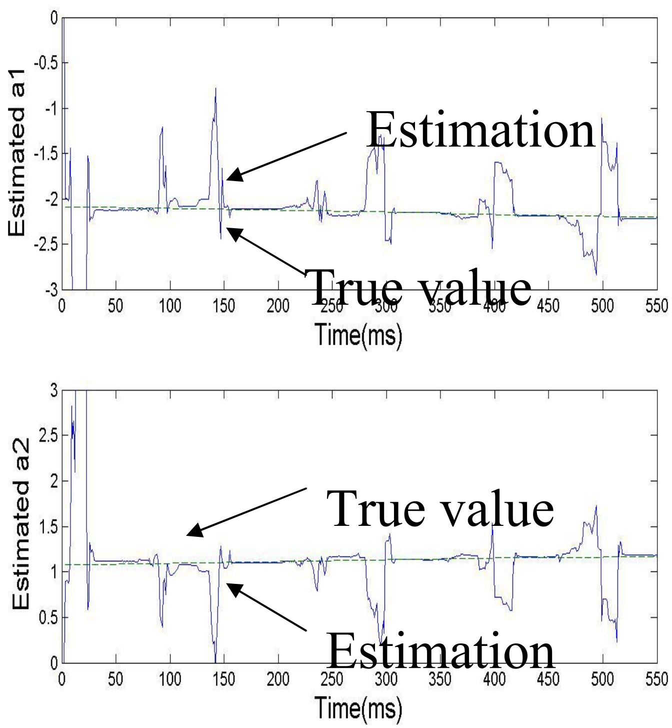
  - Relationship between λ and the evolution of the system

- Simulation of a time-varying system model
  - The simulation starts with the original model
  - The system matrix A takes step increase of 10% every 50ms:

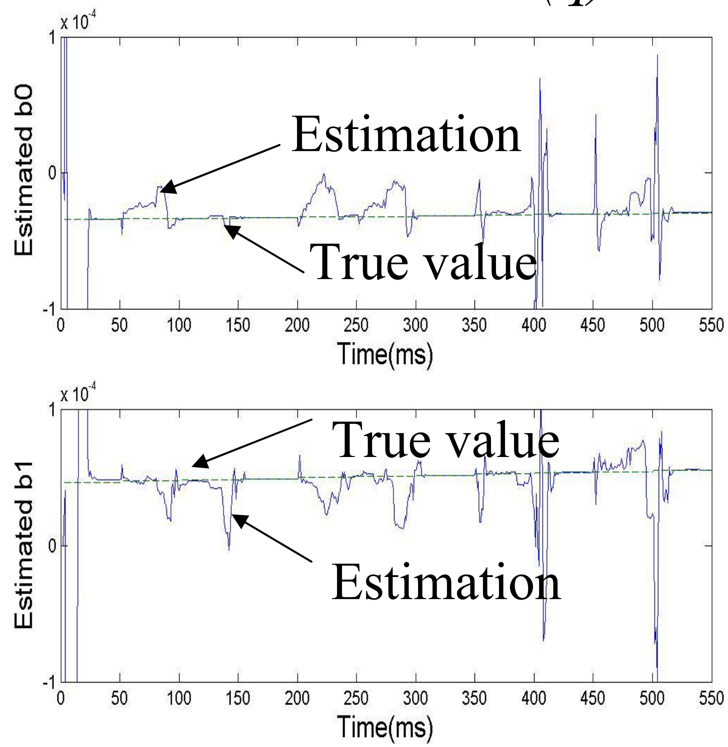
$$A \rightarrow A*1.1 \rightarrow A*1.2 \rightarrow \ldots \rightarrow A*2$$

Identification of the time-varying system

Estimation of *A(q)*



Estimation of *B(q)*



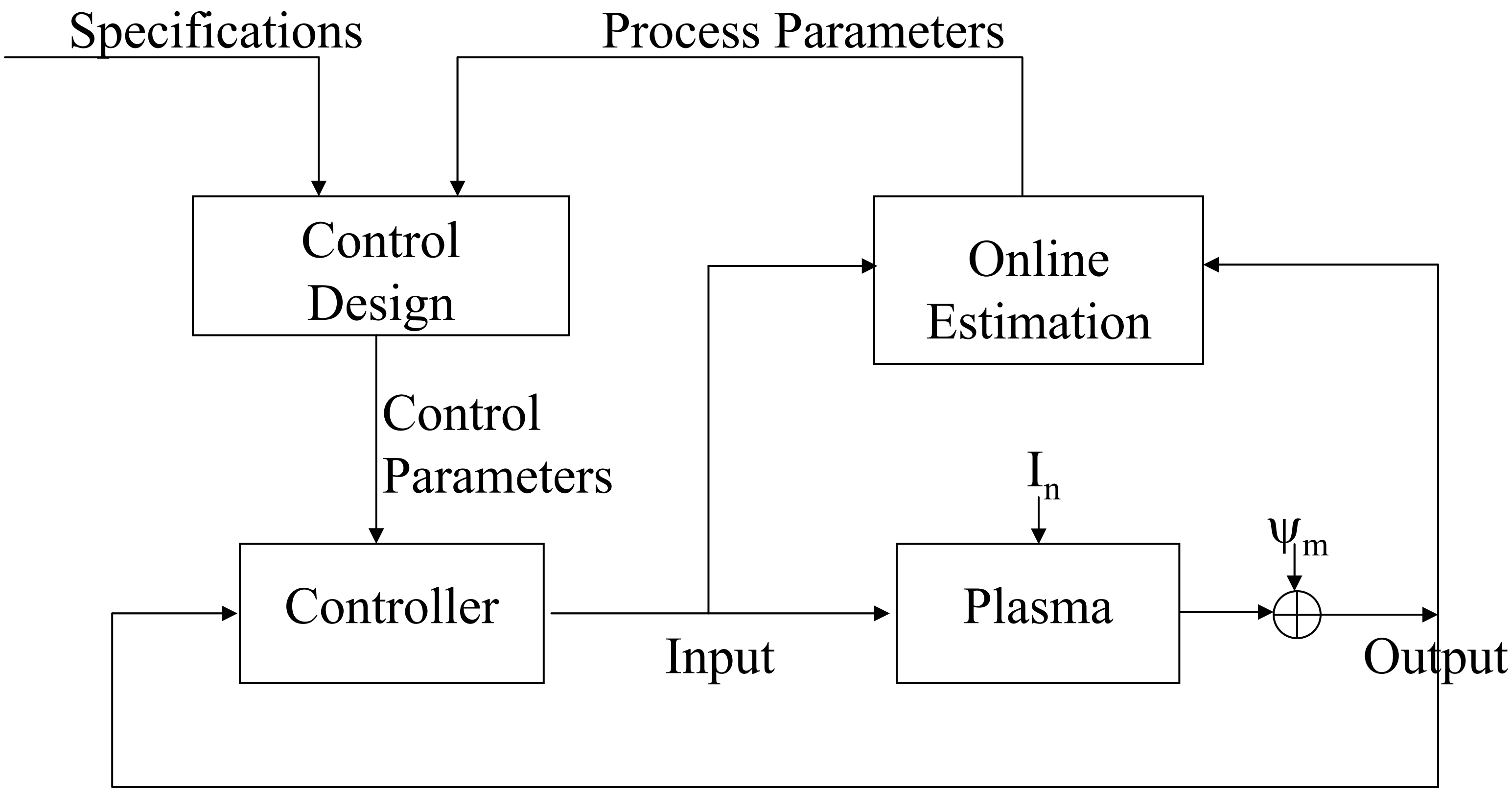
Part II

Adaptive Output Feedback Control

Objectives

- Minimize the output (fluctuation) energy and control energy
- Stabilization time is short
- Control design should be simple and fast
- Computation burden should be low

Block diagram of the controlled plasma



- Quadratic cost function

$$J = E\left\{\left(\psi(k)\right)^2 + \rho u^2\right\}$$

- Control law

$$R(q)u(q) = -S(q)\psi(q)$$

- *R* and *S* satisfy the Diophantine equation

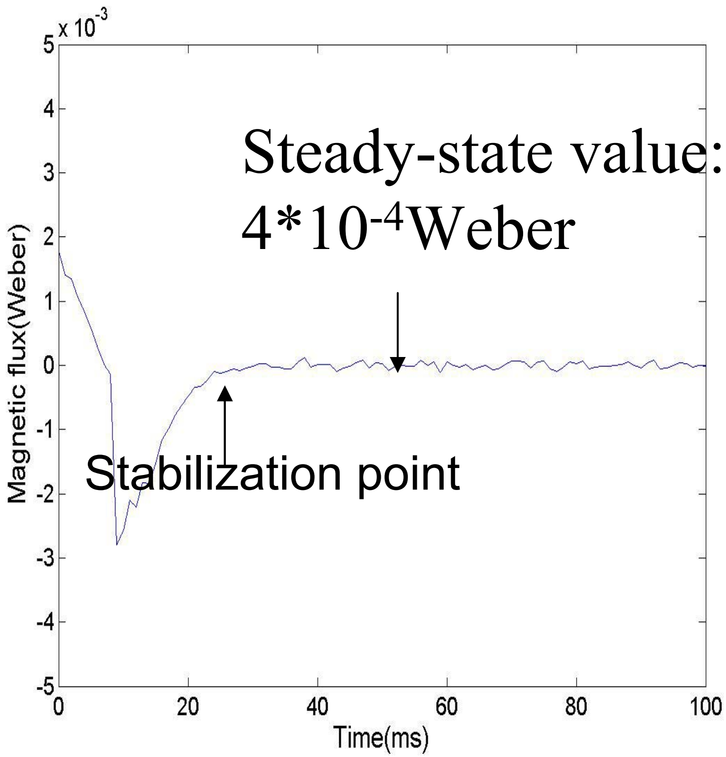
$$A(q)R(q) + B(q)S(q) = P(q)C(q)$$

- *P* is the solution to a spectral factorization problem

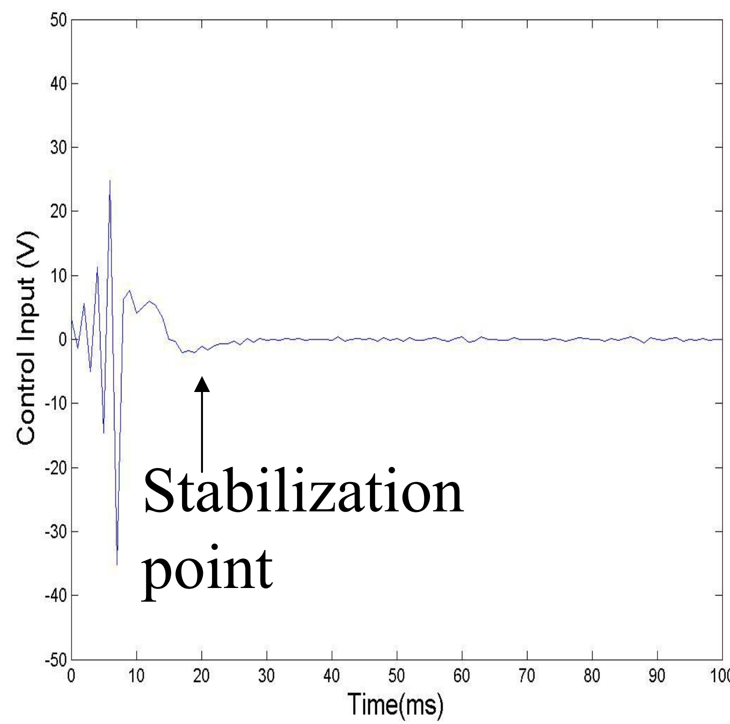
$$rP(q)P(q^{-1}) = \rho A(q)A(q^{-1}) + B(q)B(q^{-1})$$

Output feedback control of the time-invariant system

Magnetic Flux (Weber)

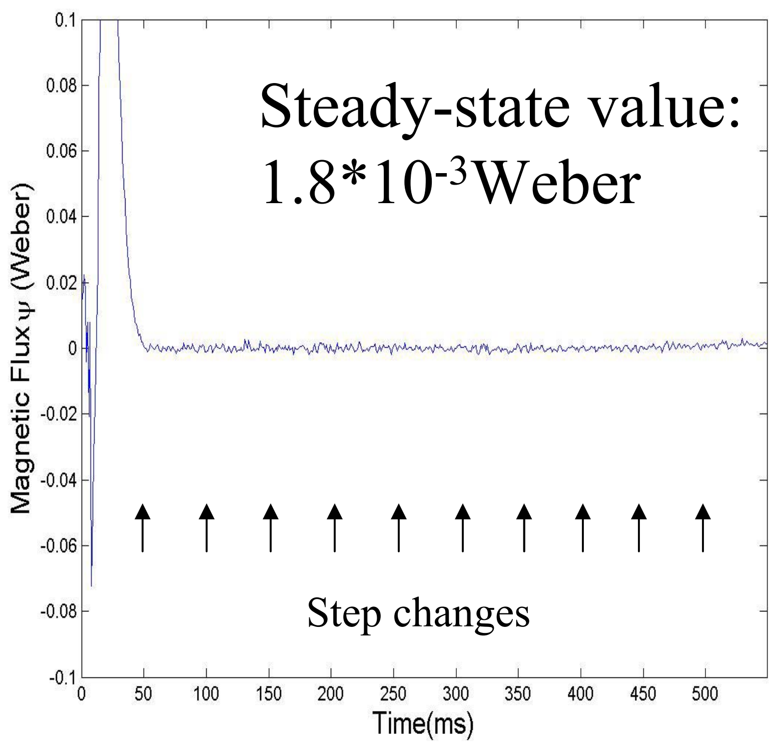


Control Signal (Volt)

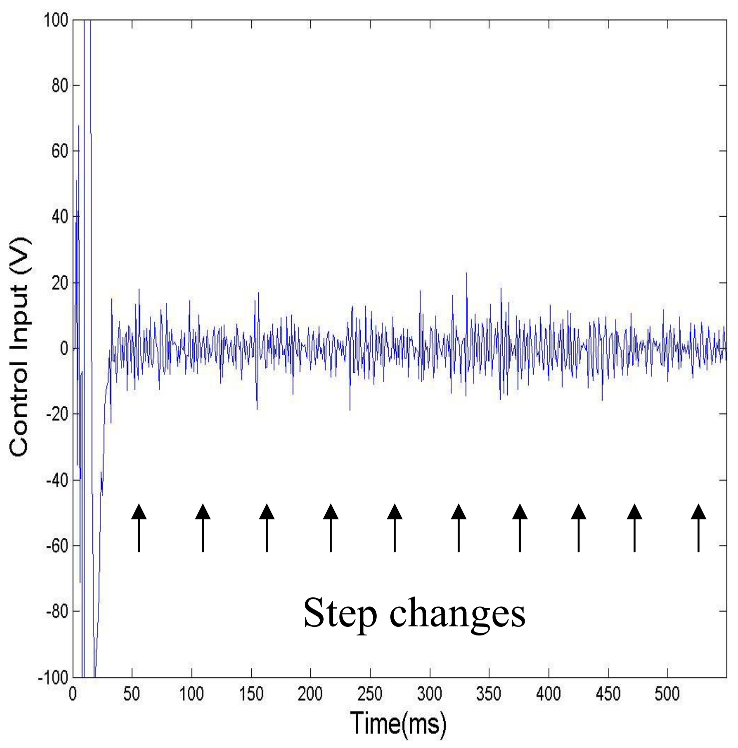


Adaptive output feedback control of the time-varying system

Magnetic Flux (Weber)



Control Signal (Volt)



Part III

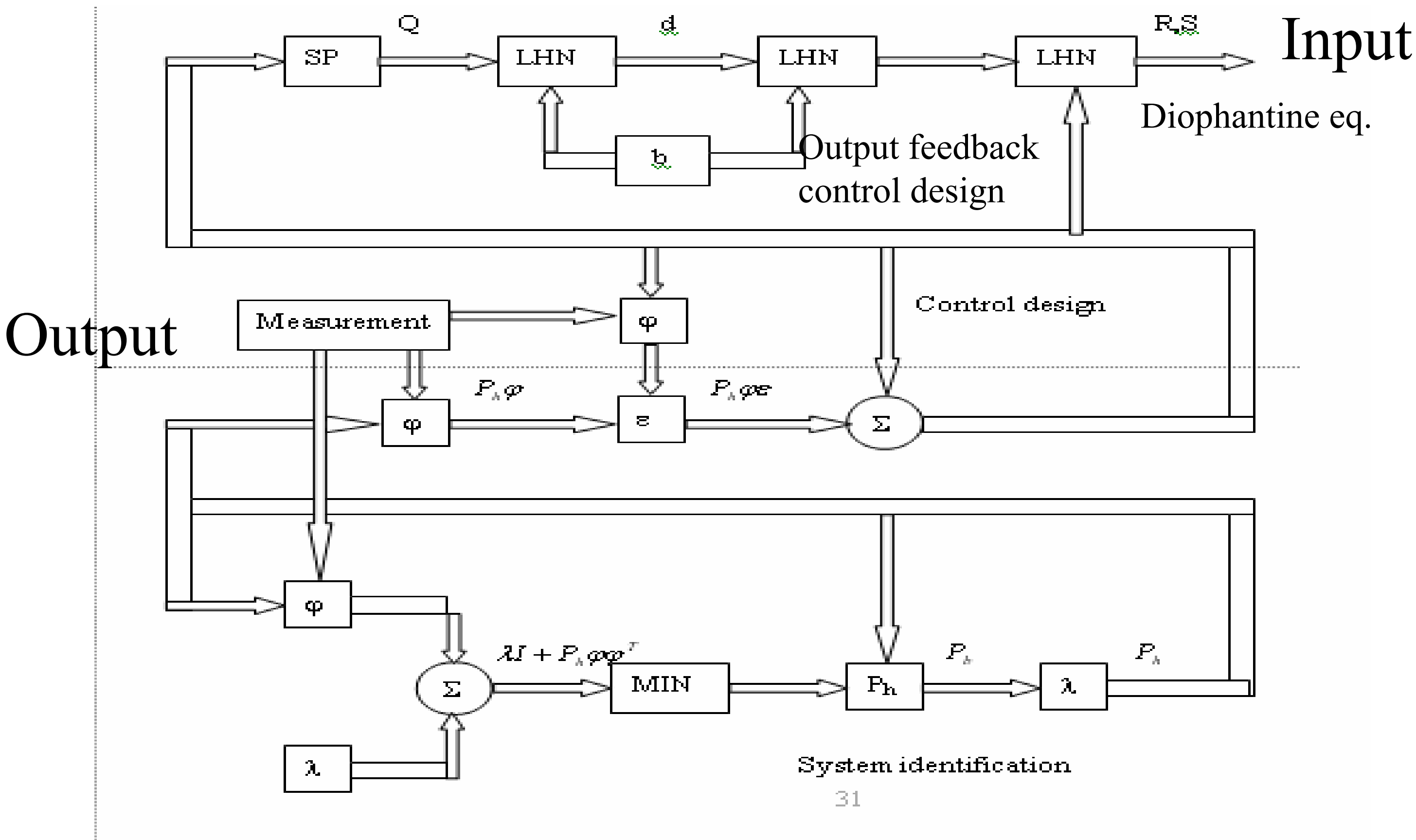
Neural Network Control

Objectives

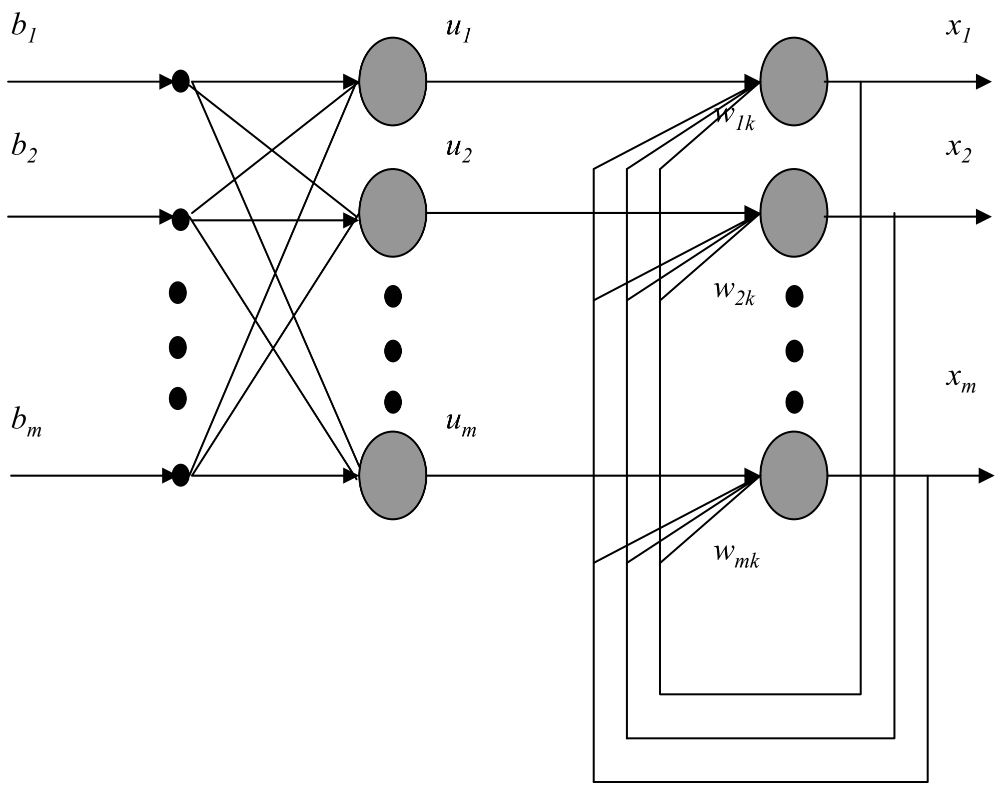
- Develop a control algorithm for a Neural Network (NN)
- Implement the adaptive output feedback control with the algorithm
- Use a digital neural network hardware

- Mathematical model for the neuron.
  - $u_1, u_2, \cdots u_m$ are inputs. They are multiplied by connection weights $w_1, w_2, \cdots w_m$ and summed.
  - The sum is passed to a transfer function and the result is the output of the neuron.
- A neural network (NN) is a system composed of many neurons
  - Its function is determined by network structure, connection strengths, and transfer functions
  - The transfer function is chosen to be a linear function in the study
- A Neural Network processor (NNP) made by Accurate Automation Corporation (AAC) has been debugged and software improved.

Block Diagram of the NNP control

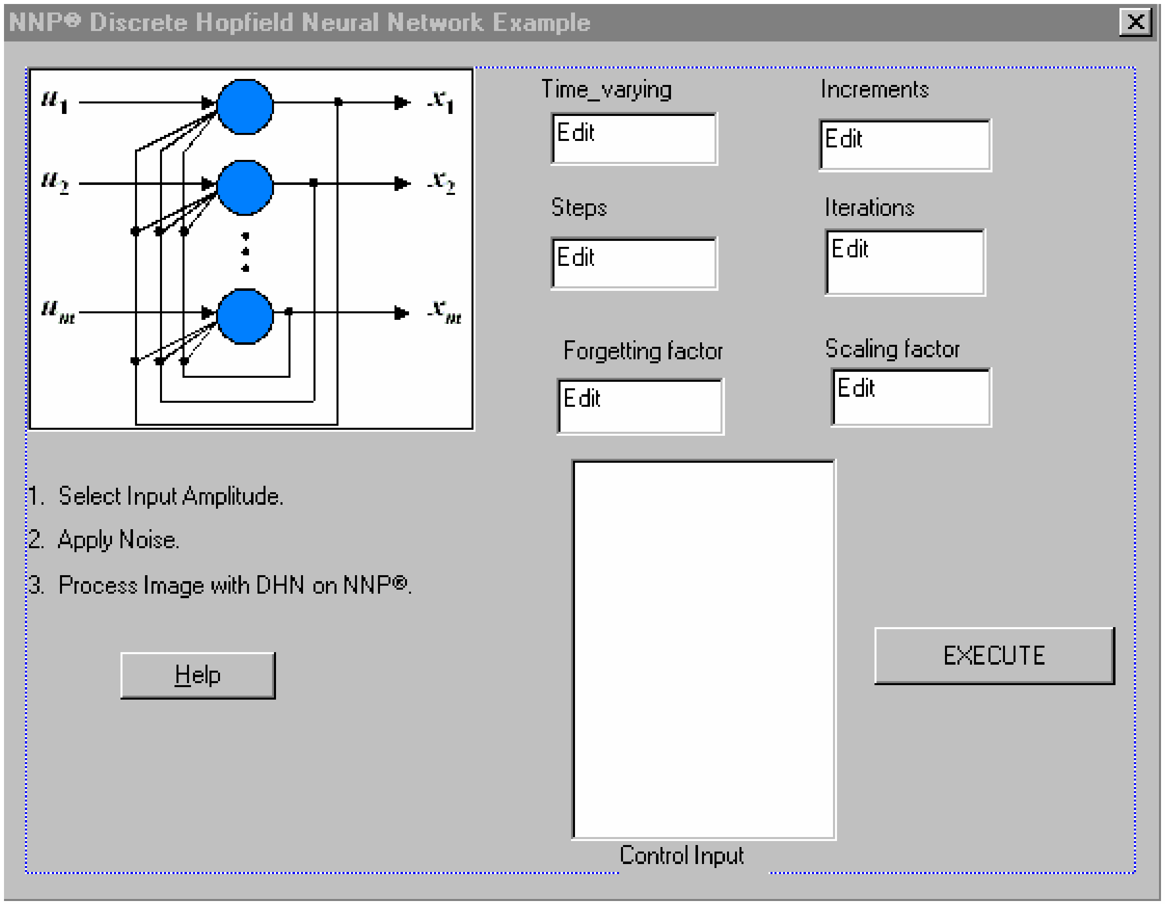


A generalized linear Hopfield network

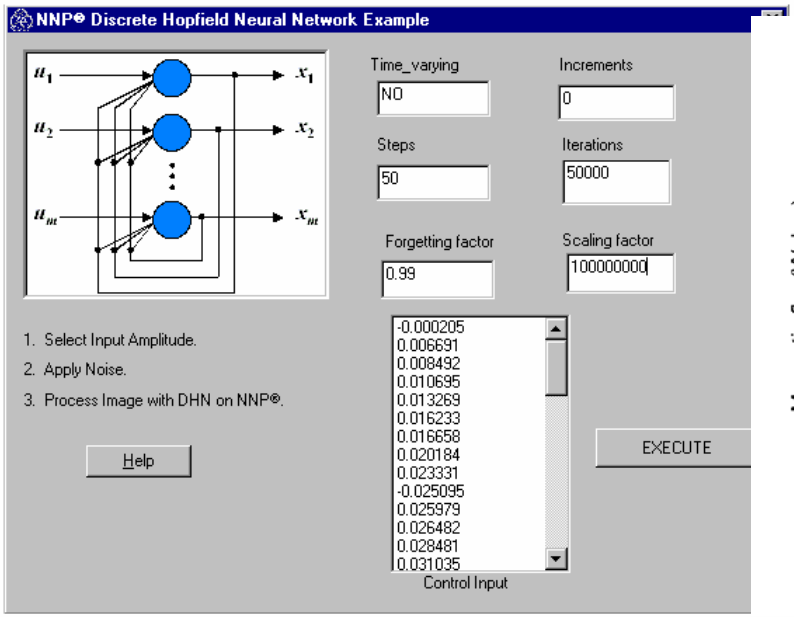


- The generalized linear Hopfield network can solve simultaneous linear equations, e.g., the Diophantine Eq.
  - The Diophantine Eq. should be rewritten as $Ax = b$
  - Stage 1: a feedforward layer with b as its inputs and A^T as its weight matrix
  - Stage 2: a Linear Hopfield layer whose inputs are the outputs of the Stage 1 layer, and weight matrix is , $W = (I - \alpha A A^T)$ where $0 < \alpha < \frac{1}{trace(A^T A)}$
  - The outputs of the second layer give the negative of the conjugate of the solution being sought.

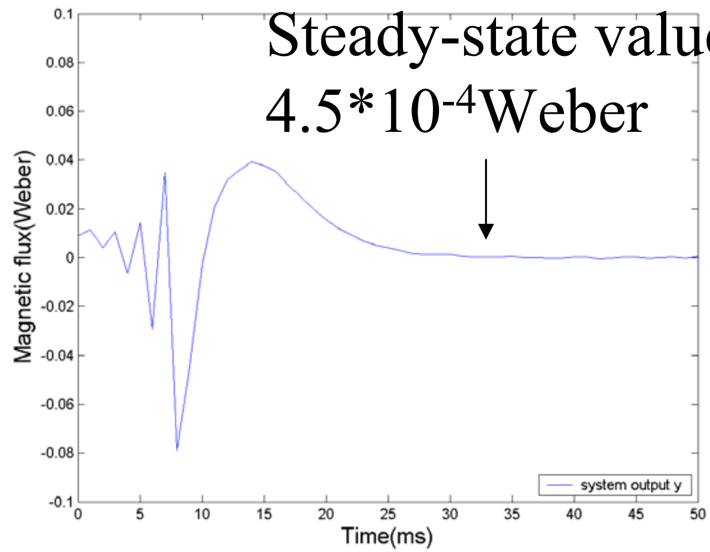
Interface of the NN controller



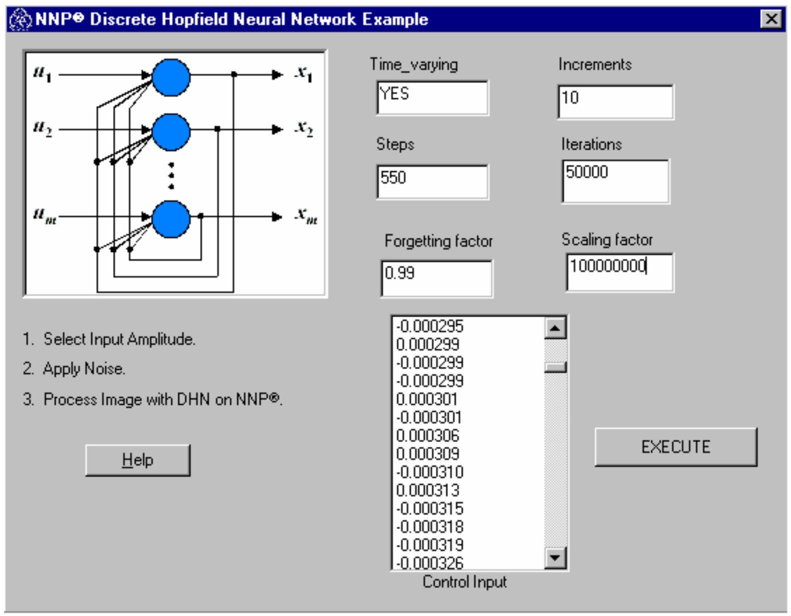
Stabilization of the time-invariant system

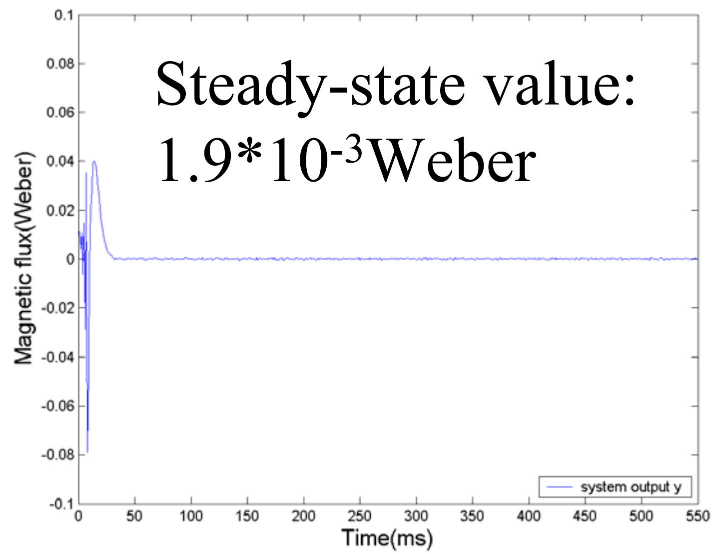


Steady-state value: $4.5*10^{-4}$Weber



Stabilization of the time-varying system





Computation time

- Matrix inversion is used as an example.
- Sequential algorithms
  - Lower-upper decomposition algorithm is used to do the inversion
  - Complexity of this algorithm is $O(N^3)$(C++ notation).
- Parallel (NN) algorithm
  - LHN is the neural network used to invert the matrix.
  - Complexity of this algorithm is either $O(N^1)$ or O(1)(C++ notation).

Summary

- The ELS method can give an accurate estimate of the single mode RWM
- Stochastic optimal output feedback control can stabilize the single mode RWM, it is able to stabilize the RWM with a convergence time of three times the inverse of the growth rate.
- Neural Network Processor can be used to implement the adaptive stochastic optimal output feedback control of a RWM.
- Computation time of the neural network control is similar to the output feedback control. However, it will be much faster for high-order systems.