Neuro-fuzzy Learning of Strategies for Optimal Control Problems*

Kaivan Kamali[1], Lijun Jiang[2], John Yen[1], K.W. Wang[2]
[1]Laboratory for Intelligent Agents
School of Information Sciences & Technology
[2]Structural Dynamics & Control Laboratory
Dept. of Mechanical & Nuclear Engineering
The Pennsylvania State University
University Park, PA 16802
Email: {kxk302,lxj148,juy1,kxw2 } @psu.edu

Abstract

Various techniques have been proposed to automate the weight selection process in optimal control problems; yet these techniques do not provide symbolic rules that can be reused. We propose a layered approach for weight selection process in which Q-learning is used for selecting weighting matrices and hybrid genetic algorithm is used for selecting optimal design variables. Our approach can solve problems that genetic algorithm alone cannot solve. More importantly, the Q-learning's optimal policy enables the training of neuro-fuzzy systems which yields reusable knowledge in the form of fuzzy if-then rules. Experimental results show that the proposed method can automate the weight selection process and the fuzzy if-then rules acquired by training a neuro-fuzzy system can solve similar weight selection problems.

1. Introduction

In traditional optimal control and design problems [6], the control gains and design parameters are derived to minimize a cost function reflecting the system performance and control effort. One major challenge of such approaches is the selection of weighting matrices in the cost function. Traditionally, selecting the weighting matrices has mostly been accomplished based on human intuition and through trial and error, which can be time consuming and ineffective. Hence, various techniques have been proposed to automate the weight selection process [1, 10, 9, 16, 2]. However, these techniques are not accompanied by a learning process which can be used to solve a similar problem.

* This research has been supported by NSF CMS grant No. 02-18597 and by AFSOR MURI grant No. F49620-00-1-0326.

We propose to model the problem of finding the optimal weighting matrices as a deterministic Markov decision process (MDP) [7]. A deterministic MDP is a MDP in which the state transitions are deterministic. Reinforcement learning (RL) techniques [11, 13, 4] such as Q-learning [15] can be used to find the optimal policy of the MDP. Modeling the problem as a deterministic MDP has the following benefits: (1) there are abundant RL techniques to solve the MDP and automate the weight selection problem, (2) the optimal policy computed by RL techniques can be used to generate fuzzy rules, which can be used in other weight selection problems. Neuro-fuzzy systems such as adaptive network-based fuzzy inference system (ANFIS) [3], can be used to learn fuzzy if-then rules for the weight selection problem using the training data obtained from RL's optimal policy.

To evaluate our method, we performed several numerical experiments on a sample active-passive hybrid vibration control problem, namely adaptive structures with active-passive hybrid piezoelectric networks (APPN) [14, 12, 5, 8]. These experiments show (1) our method can automate the weight selection problem, (2) fuzzy if-then rules are learned by training ANFIS using the training data acquired from RL's optimal policy, and (3) the learned fuzzy if-then rules can be used to solve other weight selection problems.

The rest of this paper is organized as follows: Section 2 discusses the proposed methodology. Section 3 explains a sample problem, namely, active-passive hybrid piezoelectric networks (APPN) for structural vibration control. Section 4 discusses the experimental results and Section 5 concludes the paper.

2. Proposed Methodology

In this section we first describe Markov decision processes and Q-learning. We then use an

active-passive hybrid control design problem to illustrate and evaluate our method. Finally, we provide some background information on ANFIS.

2.1. Markov Decision Processes and Q-Learning

The problem of finding a strategy for adjusting weighting matrices can be viewed as a deterministic Markov Decision Process (MDP). A Markov Decision Process consists of a set of states, S, a set of actions, A, a reward function, $R : S \times A \rightarrow \Re$, and a state transition function, $T : S \times A \rightarrow \Pi(S)$, where $\Pi(S)$ is a probability distribution over S. A deterministic MDP is a MDP in which the state transitions are deterministic. The action at each state is selected according to a policy function, $\pi : S \rightarrow A$, which maps states to actions. The goal is to find the optimal policy, π^*, which maximizes a performance metric, e.g. the expected discounted sum of rewards received.

Optimal policy of a MDP can be learned using RL techniques. Q-learning is a RL technique, which learns the optimal policy of a MDP by learning a Q-function ($Q : S \times A \rightarrow \Re$) which maps state-action pairs to values. The optimal Q-value for a state-action pair, $Q^*(s, a)$, is the expected discounted reward for taking action a in state s and continuing with the optimal policy thereafter. The optimal policy is determined from the optimal Q-value

$$\pi^*(s) = argmax_a Q^*(s, a) \tag{1}$$

Q-values are updated based on the reward received. If each state-action pair is visited frequently enough then it is guaranteed that the estimate Q-value will converge to the optimal Q-value.

2.2. Description of New Methodology

Consider the following linear system with passive control parameters $p_i (i = 1, \ldots, q)$ and active control input u:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}(p_1, \ldots, p_q)\boldsymbol{x}(t) + \boldsymbol{B_1}(p_1, \ldots, p_q)\boldsymbol{u}(t) + \boldsymbol{B_2}f(t) \tag{2}$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \tag{3}$$

where $\boldsymbol{x} \in \Re^n$ is the system state vector, $\boldsymbol{u} \in \Re^m$ is the control input vector, $\boldsymbol{y} \in \Re^p$ is the output performance vector, and f is the white Gaussian noise of excitation. The matrices $\boldsymbol{A}, \boldsymbol{B}_1, \boldsymbol{B}_2$, and $\boldsymbol{C}$ are system matrix, control input matrix, disturbance input matrix, and output matrix, respectively.

For a given set of state weighting matrices $\boldsymbol{Q}$ and control weighting matrix $\boldsymbol{R}$, the optimal values of the passive control parameters p_i can be found by minimizing the cost function, which is selected to be the minimized cost function of the stochastic regulator problem [6].

$$J = Min(\int_0^\infty [\boldsymbol{x}^T(t)\boldsymbol{Q}\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{T}\boldsymbol{u}(t)]dt) \tag{4}$$

As reported in [14] and [12], the system state weighting matrix $\boldsymbol{Q}$ and control weighting matrix $\boldsymbol{R}$ determine the solution of the regulator Ricatti equation and hence the active control gains and the optimal values of the passive control parameters. Therefore, the system performances are highly dependant on the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. Our problem is now formulated as the following: Finding the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ such that the generated active-passive hybrid control with optimal passive control parameters p_i^* and optimal active control gain $\boldsymbol{K}_c$ yields a closed-loop system which satisfies some specific constraints from the system control designers.

Following the assumption made in [1, 10, 9, 16, 2], the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are restricted to be diagonal such that

$$\boldsymbol{Q} = diag[w_m^1\ w_m^2 \ldots w_m^n] \tag{5}$$

$$\boldsymbol{R} = diag[w_c^1\ w_c^2 \ldots w_c^m] \tag{6}$$

where w_m^i $(i = 1, \ldots, n)$ is the state weighting factor on the i^{th} structural mode, w_c^i $(i = 1, \ldots, m)$ is the control weighting factor on the j^{th} control input. Therefore, the goal of the optimization problem is to find the optimal value of a weighting factor vector, $\overline{w} = ({w_m}^1, \ldots, {w_m}^n, {w_c}^1, \ldots, {w_c}^m) = (w^1, \ldots, w^{m+n})$ given the optimal values of the active feedback gain vector and the passive design variables.

The weight selection problem is cast as a RL problem, where the state space is a $(n + m)$-dimensional. In our framework, we use Q-learning to solve the RL problem. For each weight vector, the optimal value of the passive design variables are computed using hybrid GA. This layered approach, i.e. Q-learning for selecting weight vector and hybrid GA for selecting optimal passive design variables, allows for solving optimization problems that cannot be solved using GA alone. Furthermore, the Q-learning's optimal policy enables the training of neuro-fuzzy systems, e.g. ANFIS, and yields reusable knowledge in the form of fuzzy if-then rules.

The proposed approach is given in Fig. 1. The first step is to initialize the state space, which includes the following: (1) determine the range of each weight variable to be optimized, and (2) discretize the range of each weight variable. Two nested loops are then executed. The inner loop represents a single search through the state space –repeated until a maximum

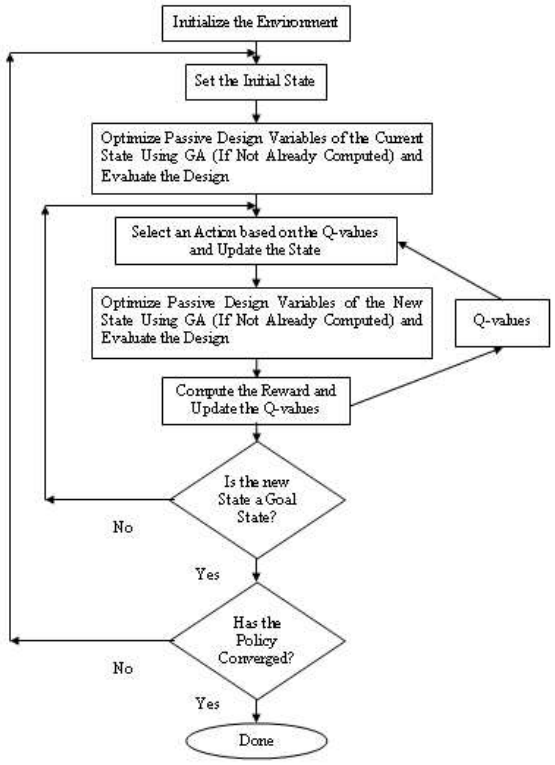

Figure 1. Overall view of the proposed layered methodology

number of iterations is reached. The outer loop is repeated until the policy for the RL agent converges.

The first step in the outer loop is to initialize the location of the RL agent in the state space, which is determined by the initial values of the elements of $\overline{w}$. This basically defines the starting point of the RL agent's search in the state space. The RL agent first finds the optimal value of the passive design variables using hybrid GA and then evaluates the design based on the design objectives. Next, it selects an action and updates its state according to the selected action.

The set of actions the RL agent can perform is denoted as $a = ({a_1}^1, {a_2}^1, \ldots, {a_1}^{n+m}, {a_2}^{n+m})$, where ${a_1}^i$ is an increase in w^i and ${a_2}^i$ is a decrease in w^i. The amount of increase or decrease Δw^i depends on how the state space is discretized. Actions are selected according to a stochastic strategy. The strategy is designed to allow for exploration of the search space by probabilistically selecting between the action corresponding to the maximum Q-value and other actions. The RL agent then finds the optimal value of the passive design variables for the new state using hybrid GA and evaluates the design using the design objectives.

In order to learn the optimal policy for weight adjustments the algorithm must compute the reward, which is calculated based on the degree in which the new design improves (or deteriorates) over the previous design. The reward is the difference between the evaluation of the current state and the previous state. Each state is evaluated by computing D, the distance between its performance (e.g. power consumption, vibration magnitude) and the design objectives. Hence, the reward is $r = D_{old} - D_{new}$. A reward is positive if the performance of the new design is closer to the design objectives.

2.3. Neuro-fuzzy Systems: ANFIS

Neuro-fuzzy systems are a combination of two popular soft computing techniques: neural networks and fuzzy systems. Neural networks have the capability to learn from examples, yet the learned knowledge cannot be represented explicitly. On the other hand, knowledge in fuzzy systems is represented via explicit fuzzy if-then rules, yet fuzzy systems have no learning capability. Neuro-fuzzy system is a hybrid approach in which a fuzzy system is trained using techniques similar to those applied to neural networks. One of the first neuro-fuzzy systems was Adaptive Network-based Fuzzy Inference System (ANFIS) [3]. ANFIS represents a Sugeno-type fuzzy system as a multilayer feedforward network which can be trained via backpropagation or a combination of backpropagation and least squares estimate.

3. An Example Problem

To evaluate the proposed method, an active-passive hybrid vibration control problem is formulated as an example for testing. This test problem is concerned with utilizing the active and passive characteristics of piezoelectric materials and circuitry for structural vibration suppression. The concept of the active-passive hybrid piezoelectric network (APPN) has been investigated by many researchers in the past [14, 12, 5, 8]. In the following, the proposed methodology is used in the design process of the active-passive hybrid piezoelectric network for vibration control of a cantilever beam.

A schematic of the system is shown in Fig. 2. A cantilever beam is partially covered with two PZT patches used as actuators. Each PZT actuator is connected to an external voltage (V_1 and V_2) source in series with a resistor-inductor shunt circuit. The beam is excited by a Gaussian white noise at $x = 0.95L_b$, and the output displacement is measured at the free-end of the beam. The two surface mounted PZT patches are located at

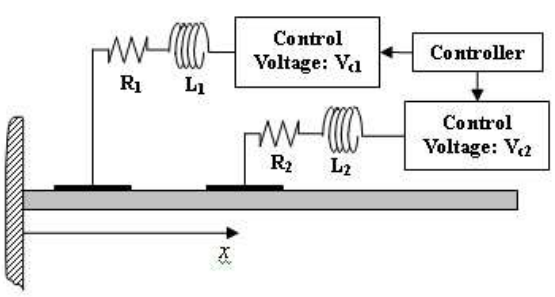

Figure 2. Schematic of the system with APPN

$x_1 = 0.02m$ and $x_1 = 0.08m$, respectively. Other parameters related to the properties of the beam and PZT patches are the same as specified in [14].

A system state space form of the system equations can be expressed as (detailed derivation of the system equations can be found in [14]):

$$\dot{x} = \boldsymbol{A}(L_1, R_1, L_2, R_2)\boldsymbol{x}(t) + \boldsymbol{B}_l(L_1, R_1, L_2, R_2)\boldsymbol{u}(t) + \boldsymbol{B}_2 f(t) \quad (7)$$

$$\boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) \quad (8)$$

$$\boldsymbol{x} = [\boldsymbol{q}^T\ \dot{\boldsymbol{q}}^T\ Q_1\ \dot{Q}_1\ Q_2\ \dot{Q}_2]^T, \boldsymbol{u}(t) = [V_{c_1}(t)\ V_{c_2}(t)]^T f(t) \quad (9)$$

where $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ are vectors of generalized displacement and velocity of the beam structure; Q_1, $\dot{Q}_1$, Q_2, and $\dot{Q}_2$ are electric charge and current of the first and second PZT actuators in the first and second piezoelectric shunting circuit, respectively; $\boldsymbol{u}$ is the control input vector composed of the control voltages V_{c_1} and V_{c_2}; $f(t)$ is the external excitation vector of Gaussian white noise. Here the system matrix A and control input matrix B_1 are function of the circuit parameters (inductances L_1, L_2 and resistances R_1, R_2).

For a given set of weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$, the optimal values of the passive design parameters (L_1, L_2, R_1, and R_2), can be found by minimizing the cost function, Eq. 4, where the system state weight matrix $\boldsymbol{Q}$ and control weigh matrix $\boldsymbol{R}$ can be expressed as

$$\boldsymbol{Q} = diag[\boldsymbol{W_Q K_b}\ \boldsymbol{W_Q M_b}\ 0\ 0\ 0\ 0] \quad (10)$$

$$\boldsymbol{W_Q} = diag[w_m^1\ w_m^2 \ldots w_m^n] \quad (11)$$

$$\boldsymbol{R} = diag[w_c^1\ w_c^2] \quad (12)$$

where $\boldsymbol{K_b}$ and $\boldsymbol{M_b}$ are the mass and stiffness matrices of the beam structure, respectively. w_c^1 and w_c^2 are scalar weighting factors on the 1^{st} and 2^{nd} control voltage source, respectively. $w_m^1\ w_m^2 \ldots w_m^n$ are scalar weighting factors on the n structural modes used in the system equation derivation, respectively. Now our design problem can be reformulated as finding the scalar weighting factors on system states ($w_m^1\ w_m^2 \ldots w_m^n$) and scalar weighting factors on control voltage sources (w_c^1, w_c^2) such that the optimal active-passive hybrid control will satisfy all constraints defined by the user.

There are different ways to define the constraints in the control system design. Eigenvalue assignment, feedback gain limit constraints, and input/output variance constraints have been used in the previous investigations. In our example constraints on the output performance and control input are defined as follows: (1) upper bounds are given to restrict the magnitudes of frequency response function at specified spatial locations and structural resonant frequencies:

$$\| G(s)\|_{s=i\omega_p} = \| \boldsymbol{C}(\boldsymbol{SI} - \boldsymbol{A} + \boldsymbol{B}_1\boldsymbol{K}_c)\boldsymbol{B}_2\|_{s=i\omega_p} \leq \mu_p\ p = 1, \ldots, n \quad (13)$$

(2) variances of the control voltage sources are bounded:

$$\lim_{x\to\infty} E[{V_{cj}}^2(t)] \leq {\sigma_j}^2 \quad j = 1, 2 \quad (14)$$

4. Experimental Results

We performed several experiments to evaluate our method. In the first experiment we used our layered approach to solve an example weight selection problem. In the second experiment we used the Q-learning's optimal policy to train ANFIS modules. We then used the trained ANFIS modules to solve two weight selection problems: the problem that ANFIS was trained with and a *different* weight selection problem.

4.1. Experiment 1

In this experiment we performed optimization on an example APPN problem. The design requirements for the APPN system are as follows:

1. Magnitude of the frequency response function (FRF) near the 1^{st} natural frequency of the system is $\leq$ -60 dB
2. Magnitude of the FRF near the 2^{nd} natural frequency of the system is $\leq$ -85 dB
3. Covariance of the 1^{st} voltage source is $\leq$ 250 V
4. Covariance of the 2^{nd} voltage source is $\leq$ 250 V

In our simulation, only the first five modes ($n = 5$) are considered. For simplicity, we also assume that (1) the two weighting factors on the control voltage source are same (i.e. $w_c^1 = w_c^2 = w_c$), and (2) the weighting factors on the structural modes other than those related to the design constraints take the value of 1 (i.e. $w_m^3 = w_m^4 = w_m^5 = 1$). Now the weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are totally determined by three design variables: w_c, w_m^1, and w_m^2. Thus the problem of finding optimal weighting matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be reformulated as finding the optimal values of the weighting factor on control voltage source (w_c) and those on

Factor	$log_{10}(w_c)$	w_m^1	w_m^1
Initial	-3	1	1
Step size	-1	0.5	0.5
Dimension	6	7	7

Table 1. Specification of the state space for experiment 1

structural modes (w_m^1 and w_m^2). The searching state space of the problem is three dimensional and specification of the state space is given in Tab. 1. The value of w_m^1 and w_m^2 increase in the state space by their corresponding step sizes, while the value of w_c is decreased by a factor which is reciprocal to its step size.

In this experiment, for each iteration of the Q-learning module, a maximum of 100 weight changes were performed. Each iteration ended when a maximum number of weight changes was performed. The program iterated until the convergence of the optimal policy. By optimal policy we mean the optimal sequence of weight changes leading to a state satisfying all the user constraints on system output and control input. The results of numerical experiment 1 are given in Tab. 2. The weights returned by the program accomplish an optimal system which satisfy all the constraints on the output performance and control effort. Our proposed method allows for automation of the optimization process, frees the expert from the burden of finding the correct sequence of weight changes, and produces reasonable results.

4.2. Experiment 2

In experiment 2, we used the Q-learning's optimal policy–learned in experiment 1–to train three ANFIS modules. The ANFIS modules are used to make weight changing decisions on w_c, w_m^1 and w_m^2. The overall methodology is given in Fig. 3. Next we describe how to get the training data for ANFIS modules.

The optimal policy is a sequence of actions (i.e. weight changes) which is derived from the Q-values. Basically, in each state the action corresponding to the maximum Q-value is the optimal action. We derived the optimal actions for each state in the state-space of experiment 1 problem. The actions corresponding to weight changes for w_c (either increase or decrease) were used as the training data for the ANFIS module corresponding to weight changing decisions for w_c. Training data for the other two ANFIS was derived similarly.

Each ANFIS module has 4 inputs: (1) magnitude of FRF near 1^{st} natural frequency, (2) magnitude of FRF near 2^{nd} natural frequency, (3) variance of the 1^{st} control voltage, and (4) variance of the 2^{nd} control voltage. Each ANFIS input has 3 membership functions associated with linguistic values *bad, average*, and *good*. The membership functions are Gaussian. Since there are 4 inputs and each input has 3 linguistic values, then each ANFIS has $3^4 = 81$ fuzzy if-then rules. ANFIS represents a Sugeno-type fuzzy inference system and the consequent for each rule is a 0^{th} degree function of the input variables.

Weighting factor on control (w_c): 1.0E-7.0
Weighting factor on the 1^{st} mode (w_m^1): 1.0
Weighting factor on the 2^{nd} mode (w_m^2): 1.0
Number of iterations for convergence: 34
Number of weight changes per iteration: 100
Magnitude of FRF near 1^{st} natural freq.: -61.5 dB
Magnitude of FRF near 2^{nd} natural freq.: -87.8 dB
Variance of the 1^{st} control voltage: 189.9 V
Variance of the 2^{nd} control voltage: 178.1 V

Table 2. Results for experiment 1

We trained the three ANFIS using a hybrid method – a combination of backpropagation and least squares estimate. After training, the three ANFIS were used to make decisions on weight selection problem for which they were trained with. Starting from the initial weight settings, hybrid GA was used to find the optimal design solution. The design solution was input to the ANFIS modules. The action corresponding to the ANFIS with the strongest output was selected. This process continued until a goal state was reached. Starting from the initial weight settings and following the procedure just discussed, the goal state was reached after 24 weight changes. The layered approach used in experiment 1 requires 3400 weight changes to solve the same problem (34 iteration for Q-learning to convergence, 100 weight changes per iteration, Tab. 2).

Next, we applied the trained ANFIS to solving a different weight selection problem. The requirements for this test problem are given in Tab. 3. Starting from the initial weight settings and following the procedure just discussed, the goal state was reached after 7 weight changes. The results are given in Tab. 4. The layered approach requires 300 weight changes to solve the same problem (3 iteration for Q-learning to convergence, 100 weight changes per iteration).

5. Summary

We propose a layered approach for solving optimal control and design problems. Such layered approach

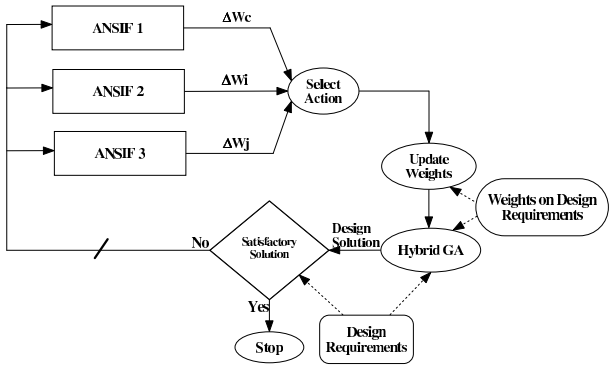

Figure 3. Overall view of weight changing methodology using ANFIS

Magnitude of FRF near 1^{st} natural freq.: -60 dB
Magnitude of FRF near 2^{nd} natural freq.: -75 dB
Variance of the 1^{st} control voltage: 200 V
Variance of the 2^{nd} control voltage: 200 V

Table 3. Test problem requirements

–i.e. Q-learning for selecting weighting matrices and hybrid GA for selecting optimal design variables– allows for solving optimization problems that cannot be solved using GA alone. Furthermore, the Q-learning's optimal policy enables the training of neuro-fuzzy systems, e.g. ANFIS, and yields reusable knowledge in the form of fuzzy rules.

To evaluate our methodology, we performed several experiments. The experiments showed that our method can successfully automate the weight selection problem. Furthermore, the Q-learning's optimal policy provides training data for ANFIS modules. ANFIS modules provide reusable fuzzy rules for the weight selection problem, which can also be applied to other weight selection problems. Moreover, the fuzzy rules provide heuristics about adjusting weights such that an acceptable design is reached much faster than using Q-learning.

Weighting factor on control (w_c): 1.0E-6.0
Weighting factor on the 1^{st} mode (w_m^1): 3.0
Weighting factor on the 2^{nd} mode (w_m^2): 1.0
Magnitude of FRF near 1^{st} natural freq.: -61.4 dB
Magnitude of FRF near 2^{nd} natural freq.: -78.0 dB
Variance of the 1^{st} control voltage: 174.8 V
Variance of the 2^{nd} control voltage: 102.3 V

Table 4. Test problem results

References

[1] A. Arar, M. Sawan, and R. Rob. Design of optimal control systems with eigenvalue placement in a specified region. *Optimal Control Applications and Methods*, 16(2):149–154, 1995.

[2] E. Collins and M. Selekwa. Fuzzy quadratic weights for variance constrained lqg design. In *Proc. of the IEEE Conference on Decision and Control*, volume **4**, pages 4044–4049, 1999.

[3] J. Jang. Anfis: Adaptive-network-based fuzzy inference systems. *IEEE Tran. on SMC*, 23(5):665–685, 1993.

[4] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *J. of AI Research*, 4(1):237–285, 1996.

[5] S. Kahn and K. Wang. Structural vibration control via piezoelectric materials with activepassive hybrid networks. In *Proc. ASME IMECE DE*, volume **75**, pages 187–94, 1999.

[6] H. Kwakernaak and R. Sivan. *Linear Optimal Control Systems*. Wiley, New York, NY, 1972.

[7] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York, NY, 1994.

[8] A. Singh and D. Pines. Active/passive reduction of vibration of periodic one-dimensional structures using piezoelectric actuators. *Smart Materials and Structures*, 13(4):698–711, 2004.

[9] B. Stuckman and P. Stuckman. Optimal selection of weighting matrices in integrated design of structures/controls. *Computers and Electrical Engineering*, 19(1):9–18, 1993.

[10] M. Sunar and S. Rao. Optimal selection of weighting matrices in integrated design of structures/controls. *AIAA J.*, 31(4):714–720, 1993.

[11] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[12] J. Tang and K. Wang. Active-passive hybrid piezoelectric networks for vibration control: Comparisons and improvement. *Smart Materials and Structures*, 10(4):794–806, 2001.

[13] G. Tesauro. Temporal difference learning and td-gammon. *Comm. of the ACM*, 38(3):58–68, 1995.

[14] M. Tsai and K. Wang. On the structural damping characteristics of active piezoelectric actuators with passive shunt. *J. of Sound and Vibration*, 221(1):1–20, 1999.

[15] C. Watkins. *Learning with delayed rewards*. PhD Thesis, Cambridge University, Cambridge, England, 1989.

[16] L. Zhang and J. Mao. An approach for selecting the weighting matrices of lq optimal controller design based on genetic algorithms. In *Proc. of IEEE Conf. on Decision and Control*, volume **3**, pages 1331–1334, 2002.