JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 30, No. 5, September–October 2007

Engineering Notes

ENGINEERING NOTES are short manuscripts describing new developments or important results of a preliminary nature. These Notes should not exceed 2500 words (where a figure or table counts as 200 words). Following informal review by the Editors, they may be published within a few months of the date of receipt. Style requirements are the same as for regular contributions (see inside back cover).

Neural Optimal Magnetohydrodynamic Control of Hypersonic Flows

Nilesh V. Kulkarni*
Perot Systems, Inc.,
Moffett Field, California 94035
and
Minh Q. Phan†
Dartmouth College, Hanover, New Hampshire 03755

DOI: 10.2514/1.26315

I. Introduction

MAGNETOHYDRODYNAMICS (MHD) as a flow control mechanism is being actively investigated for application in hypersonic systems. Several proposals suggest the use of MHD as an integral part of the flight systems [1–4]. These range from using MHD for the modification of the external flow field ahead of the flight vehicle to internal flow control for aircraft engine application. Our previous work developed an open-loop optimal control approach for an MHD channel at the inlet of an air-breathing hypersonic propulsion system [5]. That approach maximizes the performance of this MHD channel by specifying a profile for the electron beam current along the channel by assuming the knowledge of the flow variables at the lip of the channel. The results show that the flow profile can be successfully controlled for maximizing the net energy extracted from the system. The flow Mach number at the channel exit can be closely held to a prespecified value for a range of inlet conditions. This can make the scramjet combustor design much easier, as it does not have to account for a range of flow Mach numbers. In this work, we assume that the knowledge of system states is possible at a few positions in the channel through the use of sensors at these locations. We then propose a mixed predictive control and dynamic-programming-based control design that provides the optimal e-beam current settings along the particular portion of the channel based on the state information from the adjacent sensor.

Section II describes the hypersonic MHD channel, and presents the steady-state flow equations used in the simulation. Section III outlines the mixed predictive control and dynamic-programming-based feedback optimal control approach, as applied to the MHD channel. Section IV presents the results of the neural network training and optimization. Finally, Sec. V outlines the conclusions of the work, and suggests directions for future research.

II. System Description

The system under consideration is a hypersonic channel at the inlet of the MHD energy bypass engine. The flow enters the channel (in the x direction), where it is subjected to ionizing electron beams. As shown in Fig. 1, the electron beams are introduced in the same direction (y direction) as the applied magnetic field. As the ionized flow passes across the magnetic field lines, an electromotive force (emf) is generated (in the z direction) perpendicular to both the flow direction and the direction of the applied magnetic field. This physical effect is based on Faraday's law of electromagnetic induction [4]. The induced current represents electrical energy that can be extracted from the flow. In the process, the flow experiences a $\boldsymbol{j}_y \times \boldsymbol{B}$ force that decelerates the flow. In this configuration, the channel is assumed to be an ideal Faraday generator [4]. The geometry of the hypersonic inlet channel in this study is chosen to be consistent with the one used in [6–8].

Specifications for the MHD Channel:

1) Channel length, 1.6 m.
2) Inlet area, 0.0225 m^2 (15 cm × 15 cm).
3) Exit area, 0.1225 m^2 (35 cm × 35 cm).
4) The width of the channel is assumed to vary linearly along the length of the channel.
5) Applied external magnetic field, 5 T.
6) Freestream altitude, 30 km (±10%).
7) Freestream Mach number, 8 (±10%).

The governing equations assume one-dimensional steady-state flow with the added MHD terms. The system dynamics can be described as

$$\frac{d\boldsymbol{w}}{dx} = \boldsymbol{f}[\boldsymbol{w}(x), \boldsymbol{u}(x), x] \tag{1}$$

Here $\boldsymbol{u}(x)$ corresponds to the control, which in this case is the electron beam current, and x corresponds to the position variable. $\boldsymbol{w}(x)$ corresponds to the state vector for the system, which is given as

$$\boldsymbol{w}(x) = \begin{bmatrix} \rho_f & v_f & P_f & n_{ef} \end{bmatrix}^T \tag{2}$$

ρ_f (kg/m^3) is the density of the flow, v_f (m/s) is the velocity of the fluid, P_f (N/m^2) is the static pressure of the fluid, and n_{ef} (1/m^3) is the electron number density along the channel. Reference [5] provides more details on this MHD model.

In supersonic flow, effects of inputs given to the flow at a given location are only felt downstream of that location. The x coordinate along the flow therefore behaves like the time coordinate, as any event occurring at time t affects the system only at later times. Optimal control theory that has been extensively developed with time as the independent variable can, therefore, be applied to the MHD channel that has position as the independent variable.

III. Dynamic-Programming-Based Optimal Control Design

The structure of the sensor and actuator placement, as outlined in Fig. 1, motivated a mixed predictive control and dynamic-programming-based off-line optimal control design. Reference [9]

Presented as Paper 5497 at the AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, 11–14 August 2003; received 3 July 2006; revision received 19 February 2007; accepted for publication 12 March 2007.

*Scientist, NASA Ames Research Center; nilesh@email.arc.nasa.gov. Member AIAA.
†Associate Professor, Thayer School of Engineering; Minh.Q.Phan@dartmouth.edu. Member AIAA.

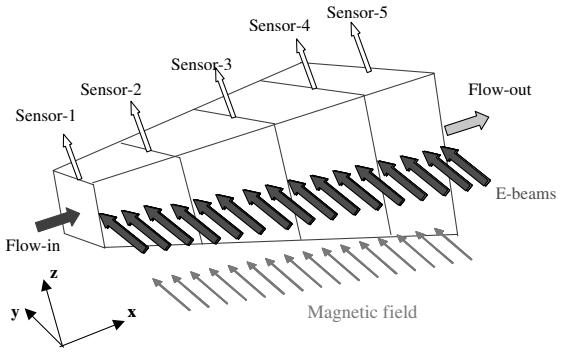

Fig. 1 Schematic of the MHD power generator.

describes the general nature of this parameterized predictive control design. The system states between sensor locations are predicted using trained neural network models. These are used in a dynamic-programming-based architecture, which designs the four controllers, starting with the last one first and moving upstream till the inlet controller. The control design uses two groups of neural networks. The first group is used to model the controller, and the second group is used to model the cost-to-go function.

A. Neural Network Controller

The optimal controller corresponds to a feedback function of the sensed state. There are only a discrete number of sensors in the channel. Based on the sensed state at each sensor location, the optimal controller, therefore, provides the control inputs from that sensor location to the next sensor location. The controller design is correspondingly broken down into the design of four controllers, one each for the first four sensors. Each of the controllers is parameterized using neural networks.

B. Neural Network Cost-to-Go Function Estimator

The cost-to-go function, $V[\boldsymbol{w}(x_c), \bar{\boldsymbol{u}}(x_c, x_f), x_c]$, is defined as

$$V[\boldsymbol{w}(x_c), \bar{\boldsymbol{u}}(x_c, x_f), x_c] = p_1[M_f(x_f) - M_{fe}]^2 + \int_{x_c}^{x_f} \left[\frac{q_1}{\rho_f v_f A_f} \left[Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f \right] + r_1 j_{bf}^2 \right] \mathrm{d}x \tag{3}$$

$\bar{\boldsymbol{u}}(x_c, x_f)$ denotes the control profile from the position x_c to the end position x_f. Contrary to the cost function J, $V[\boldsymbol{w}(x_c), \bar{\boldsymbol{u}}(x_c, x_f), x_c]$ is defined for every state $\boldsymbol{w}(x)$, at all positions. Minimizing the cost-to-go function, therefore, provides a feedback controller, which gives the optimal control profile as a function of the system state $\boldsymbol{w}(x)$. The different terms in this cost-to-go function achieve the following:

1) To minimize deviations from prescribed flow Mach number at the end of the channel.

2) To maximize the net energy extracted from the system that corresponds to the difference between the energy extracted from the flow and the energy spent on the *e*-beam ionization.

3) To minimize the net usage of the *e*-beam current.

The principle of dynamic programming is used to compute the optimal controllers at the first four sensor locations. Bellman and his colleagues formulated the theory of dynamic programming toward finding optimal solutions for multistage sequential decision-making problems [10]. The essence of the dynamic programming solution procedure lies in solving the optimization problem in a backward manner. Given a multistage problem, dynamic programming calculates the optimal decisions for the states of the system at the last-but-one stage. It then uses this result to compute the optimal decisions for the system states at the last-but-two stage, and this procedure is repeated in a backward manner till the first stage. The current problem, equivalently, corresponds to a five-stage sequential decision-making problem. The decisions correspond to the control inputs given to the flow based on the information at the corresponding sensor locations. The fifth stage corresponds to the end position that sees the effect of the *decisions* in the final quarter section of the channel.

With this understanding, four cost-to-go function networks are designed for the first four sensor locations, which predict the cost-to-go function from that sensor location to the end of the channel. The solution procedure starts at the fourth sensor location. The cost-to-go function estimator (CFE) network for this sensor location is designed as outlined in [5]. The controller network outputs are attached to the CFE network control inputs to form a combined network. Figure 2 illustrates this combined network. The combined network takes the state of the system, $\boldsymbol{w}(x_{\text{sensor4}})$, as its input and gives the cost-to-go function estimate as its output. This corresponds to the equation

$$V\{\boldsymbol{w}(x_{\text{sensor4}}), \boldsymbol{g}_{\text{NN}}[\boldsymbol{w}(x_{\text{sensor4}})], x_{\text{sensor4}}\} = p_1[M_f(x_f) - M_{fe}]^2 + \int_{x_{\text{sensor4}}}^{x_f} \left[\frac{q_1}{\rho_f v_f A_f} \left[Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f \right] + r_1 j_{bf}^2 \right] \mathrm{d}x \tag{4}$$

$$\bar{\boldsymbol{u}}(x_{\text{sensor4}}, x_f) = \boldsymbol{g}_{\text{NN}}[\boldsymbol{w}(x_{\text{sensor4}})] \tag{5}$$

$\boldsymbol{g}_{\text{NN}}[\boldsymbol{w}(x_{\text{sensor4}})]$ represents the neural network controller part of the combined network that produces the control profile from the sensor 4 location to the channel exit. To optimize the neural network controller, the combined network is trained as a whole. However, the weights and biases corresponding to the CFE part of the network are held fixed. For a randomized set of the achievable states at the sensor 4 location, the controller part of the network is updated to minimize the output of the combined network. Figure 3 illustrates this approach.

After optimizing the sensor 4 controller, the dynamic programming principle is used for computing the optimal controllers for the previous sensor locations. The dynamic programming principle for this MHD implementation can be stated as follows: the

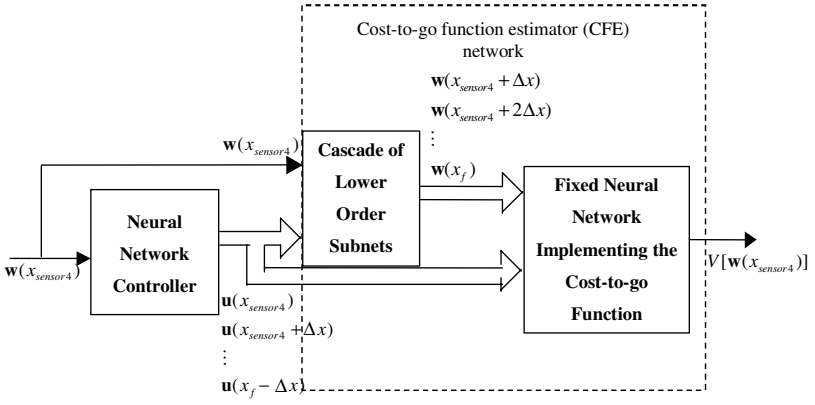

Fig. 2 Sensor 4 combined network consisting of the neural network controller and the CFE network.

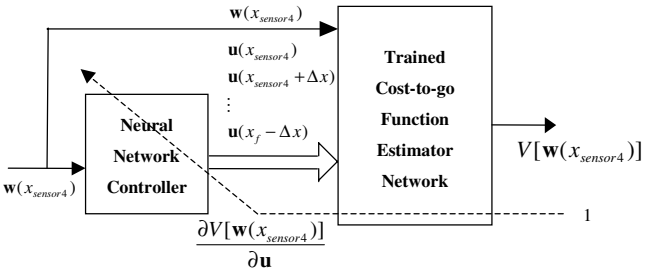

Fig. 3 Training the controller part of the sensor 4 combined network.

optimal policy has the property that whatever the initial state is at sensor location 3 and whatever the control settings are from sensor 3 to sensor 4, the optimal controller designed from sensor 4 to the end of the channel remains the same based on the state generated at sensor location 4 due to the state at sensor location 3 and the control actuation from sensor 3 to sensor 4.

The cost-to-go function from the third sensor is given as

$$V[\boldsymbol{w}(x_{\text{sensor3}}), \bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_f), x_{\text{sensor3}}] = p_1[M_f(x_f) - M_{fe}]^2 + \int_{x_{\text{sensor3}}}^{x_f} \left[\frac{q_1}{\rho_f v_f A_f} \left[Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f \right] + r_1 j_{bf}^2 \right] \mathrm{d}x \tag{6}$$

To minimize $V[\boldsymbol{w}(x_{\text{sensor3}}), \bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_f), x_{\text{sensor3}}]$, the control profile from the location of sensor 3 to the end of the channel needs to be optimized. However, $V[\boldsymbol{w}(x_{\text{sensor3}}), \bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_f), x_{\text{sensor3}}]$ can be broken as

$$V[\boldsymbol{w}(x_{\text{sensor3}}), \bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_f), x_{\text{sensor3}}] = \int_{x_{\text{sensor3}}}^{x_{\text{sensor4}}} \left[\frac{q_1}{\rho_f v_f A_f} \left[Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f \right] + r_1 j_{bf}^2 \right] \mathrm{d}x + V[\boldsymbol{w}(x_{\text{sensor4}}), \bar{\boldsymbol{u}}(x_{\text{sensor4}}, x_f), x_{\text{sensor4}}] \tag{7}$$

The optimal sensor 3 cost-to-go function can thus be given as

$$\begin{aligned} V^*[\boldsymbol{w}(x_{\text{sensor3}}), \bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_f), x_{\text{sensor3}}] &= \min_{\bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_f)} \left\{ \int_{x_{\text{sensor3}}}^{x_{\text{sensor4}}} \left[\frac{q_1}{\rho_f v_f A_f} [Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f] + r_1 j_{bf}^2 \right] \mathrm{d}x + V[\boldsymbol{w}(x_{\text{sensor4}}), \bar{\boldsymbol{u}}(x_{\text{sensor4}}, x_f), x_{\text{sensor4}}] \right\} \\ &= \min_{\bar{\boldsymbol{u}}(x_{\text{sensor3}}, x_{\text{sensor4}})} \left\{ \int_{x_{\text{sensor3}}}^{x_{\text{sensor4}}} \left[\frac{q_1}{\rho_f v_f A_f} [Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f] + r_1 j_{bf}^2 \right] \mathrm{d}x + V^*[\boldsymbol{w}(x_{\text{sensor4}}), \boldsymbol{g}_{\text{NN}}^*[\boldsymbol{w}(x_{\text{sensor4}})], x_{\text{sensor4}}] \right\} \end{aligned} \tag{8}$$

With the sensor 4 cost-to-go function and controller optimized, the dynamic programming principle says that to optimize the sensor 3 cost-to-go function, only the extra integral term

$$U(x_{\text{sensor3}}, x_{\text{sensor4}}) = \int_{x_{\text{sensor3}}}^{x_{\text{sensor4}}} \left[\frac{q_1}{\rho_f v_f A_f} \left[Q_{\beta f} A_f - k_f(1-k_f)\sigma_f v_f^2 B_f^2 A_f \right] + r_1 j_{bf}^2 \right] \mathrm{d}x \tag{9}$$

needs to be optimized. $U(x_{\text{sensor3}}, x_{\text{sensor4}})$ represents the utility function between sensor 3 and sensor 4 locations. The controller that was previously optimized to give the optimal control profile from sensor 4 to the end of the channel remains the same. Figure 4 illustrates the sensor 3 combined network, and the training approach for the sensor 3 controller.

The same procedure is repeated for optimizing the sensor 2 and sensor 1 controllers. Even though the overall procedure seems quite complex from an implementation standpoint, it is easily handled by building custom networks using the MATLAB® neural network toolbox. The principle of dynamic programming allows us to break up this problem, and train smaller control structures to give optimal control profiles for portions of the channel using information from the respective sensors.

IV. Results

The neural network multistep ahead models (subnets) are trained using the Levenberg–Marquardt algorithm of the MATLAB® neural network toolbox. For creating a rich training data, the MHD system dynamics are simulated with random inlet conditions, and random *e*-beam current values. The altitude and Mach number are chosen from a uniform distribution with a range of 10% around the design altitude

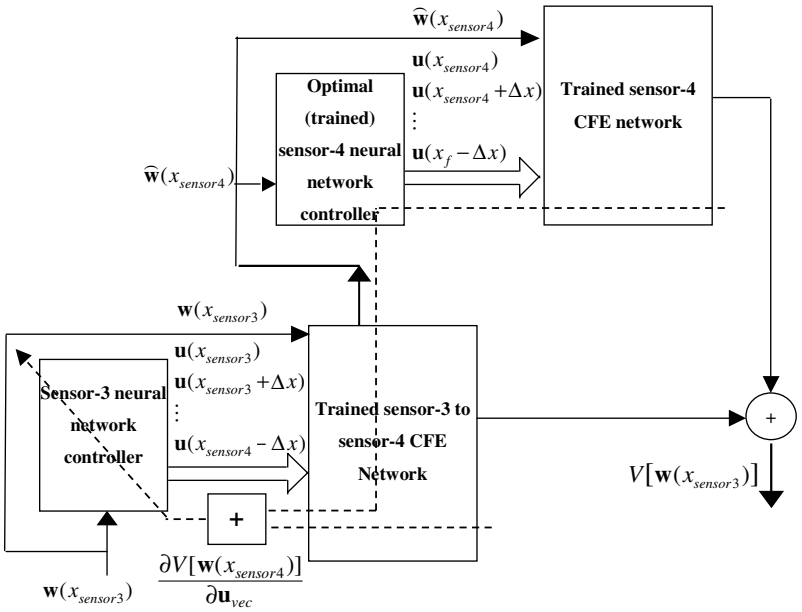

Fig. 4 Training the controller from sensor 3 to sensor 4 using the dynamic programming principle.

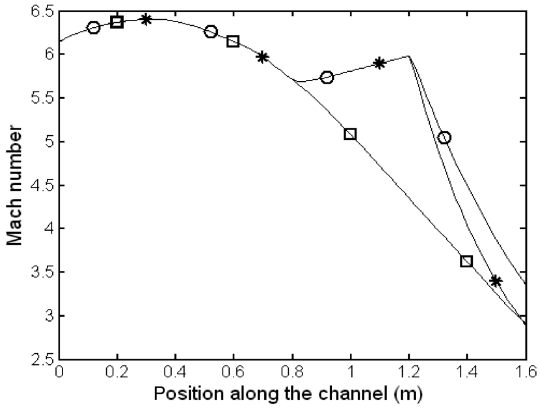

Fig. 5 Mach number profiles to illustrate feedback approach. □, with no failure; ○, open loop with failure; *, feedback with failure.

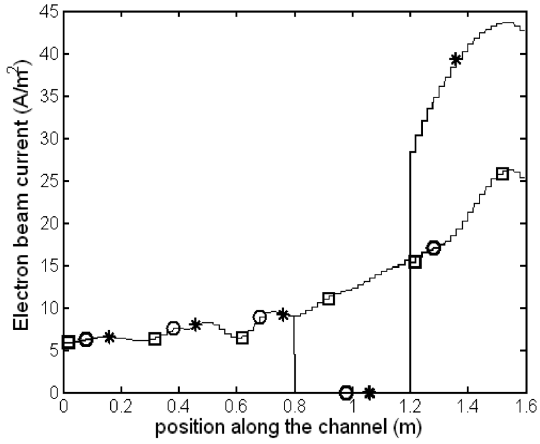

Fig. 6 Electron beam current profiles to illustrate feedback approach. □, with no failure; ○, open loop with failure; *, feedback with failure.

of 30 km and design Mach number of 8. The *e*-beam windows are assumed to allow a maximum current output of 50 A/m^2. The *e*-beam current is chosen with a uniform distribution between 0 and 50 A/m^2.

The results illustrate the feedback nature of the control architecture, as opposed to the open-loop approach designed in [5]. A more complete set of results is given in [11]. The open-loop approach assumed only a single sensor at the inlet, based on which, the control inputs are given along the entire channel. Any failures occurring in the channel, therefore, will remain undetected. For these results, we assume that the control actuators between the location of sensor 3 and sensor 4 have failed, and do not output any electron beams. The cost function prescribes an exit Mach number of 3, while maximizing the net energy extracted, and minimizing the net control usage. The weighting parameters that specify this cost function are the following: p_1, 20; q_1, 0.00001; r_1, 0.005.

Figures 5 and 6 illustrate the results for the Mach number profile and the control profile, respectively. The freestream Mach number of the flow is 8, and the flight altitude is 30 km. We look at three different trajectories. The first (□) corresponds to the no-failure case. The exit Mach number comes close to the prescribed value of 3. The second trajectory (○) represents the open-loop trajectory with all actuators failing between sensor 3 and sensor 4 locations. The open-loop nature of this trajectory is implemented by applying the same control inputs to the flow as in the no-failure case. This assumes that there is no knowledge of the failure having taken place. The exit Mach number correspondingly differs substantially from the prescribed value. The third trajectory (*) represents the same actuator failures between the sensor 3 and sensor 4 location. In this case, however, sensor 4 senses the different state variables that reflect the actuator failures. The control profile from the location of sensor 4 to the end of the channel is given by the sensor 4 controller based on these new variables. With this feedback control, the exit Mach number gets back to the prescribed value of 3.

V. Conclusions

In this work, the performance optimization of the MHD generator is treated as a feedback optimal control problem. Assuming multiple sensors available along the channel, the proposed algorithm uses the dynamic programming principle to get a feedback form of the controller. The neural network controllers successfully optimize different system performance measures that take into account factors such as energy extraction, exit Mach number, energy spent for *e*-beam ionization, and so on. The feedback nature of the control architecture is successfully illustrated for actuator failures. This design approach represents an off-line approach, which uses the existing information about the system to design the optimal controllers. In our subsequent work, we consider the application of reinforcement learning algorithms for in-flight reoptimization of these off-line controllers to account for modeling errors [11,12].

Acknowledgments

This research was supported by a grant from the National Science Foundation through ANSER Corporation. The authors acknowledge the magnetohydrodynamics research group at Princeton University under Richard Miles. In particular, we are grateful to Mikhail Shneyder for his assistance in modeling the magnetohydrodynamics channel.

References

[1] Chase, R. L., Mehta, U. B., Bogdanoff, D. W., Park, C., Lawrence, S., Aftosmis, M., Macheret, S. O., and Shneyder, M. N., "Comments on a MHD Bypass Spaceliner Performance," AIAA Paper 99-4965, Nov. 1999.

[2] Fraishtadt, V. L., Kuranov, A. L., and Sheikin, E. G., "Use of MHD Systems in Hypersonic Aircraft," *Technical Physics*, Vol. 43, No. 11, 1998, p. 1309.

[3] Gurijanov, E. P., and Harsha, P. T., "AJAX: New Directions in Hypersonic Technology," AIAA Paper 96-4609, 1996.

[4] Rosa, J. R., *Magneto-hydrodynamic Energy Conversion*, McGraw–Hill, New York, 1968.

[5] Kulkarni, N. V., and Phan, M. Q., "Performance Optimization of a Magnetohydrodynamic Generator at the Scramjet Inlet," *Journal of Propulsion and Power*, Vol. 21, No. 5, Sept.–Oct. 2005, pp. 822–830.

[6] Macheret, S. O., Schneider, M. N., Miles, R. B., and Lipinski, R. J., "Electron Beam Generated Plasmas in Hypersonic Magnetohydrodynamic Channels," *AIAA Journal*, Vol. 39, No. 6, 2001, pp. 1127–1138.

[7] Macheret, S. O., Schneider, M. N., and Miles, R. B., "MHD Power Extraction from Cold Hypersonic Air Flows with External Ionizers," AIAA Paper 99-4800, Nov. 1999.

[8] Macheret, S. O., Schneider, M. N., and Miles, R. B., "Potential Performance of Supersonic MHD Power Generators," AIAA Paper 2001-0795, Jan. 2001.

[9] Kulkarni, N. V., and Phan, M. Q., "Neural-Network-Based Design of Optimal Controllers for Nonlinear Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, Sept.–Oct., 2004, pp. 745–751.

[10] Bellman, R. E., *Dynamic Programming*, Princeton Univ. Press, Princeton, NJ, 1957.

[11] Kulkarni, N. V., "Predictive and Reinforcement Learning for Magnetohydrodynamic Control of Hypersonic Flows," Ph.D. Dissertation, Department of Mechanical and Aerospace Engineering, Princeton Univ., Princeton, NJ, 2007.

[12] Kulkarni, N. V., and Phan, M. Q., "Reinforcement-Learning-Based Magnetohydrodynamic Control of Hypersonic Flows," *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, IEEE Publications, Piscataway, NJ, 2007.