

01 Jan 2006

Dynamic Re-Optimization of a Spacecraft Attitude Controller in the Presence of Uncertainties

Nishant Unnikrishnan

S. N. Balakrishnan

Missouri University of Science and Technology, bala@mst.edu

Radhakant Padhi

Follow this and additional works at: https://scholarsmine.mst.edu/mec_aereng_facwork



Part of the [Aerospace Engineering Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

N. Unnikrishnan et al., "Dynamic Re-Optimization of a Spacecraft Attitude Controller in the Presence of Uncertainties," *Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.

The definitive version is available at <https://doi.org/10.1109/CACSD-CCA-ISIC.2006.4776688>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Mechanical and Aerospace Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

Dynamic Re-optimization of a Spacecraft Attitude Controller in the Presence of Uncertainties

Nishant Unnikrishnan, S. N. Balakrishnan, and Radhakant Padhi,

Abstract—Online trained neural networks have become popular in recent years in the design of robust and adaptive controllers for dynamic systems with uncertainties due to their universal function approximation capabilities. This paper discusses a technique that dynamically reoptimizes a Single Network Adaptive Critic (SNAC) based optimal controller in the presence of unmodeled plant uncertainties. The SNAC based optimal controller designed for the nominal plant model no more retains optimality in the presence of uncertainties/unmodeled dynamics that may creep up in the system equations during operation. This calls for a strategy to re-optimize the existing SNAC controller with respect to the original cost function but corresponding to new constraint (state) equations. The controller re-optimization is carried out in two steps: (i) synthesis of a set of online neural networks that capture the uncertainties in the plant equations on-line (ii) re-optimization of the existing SNAC controller to drive the states of the plant to a desired reference by minimizing the original cost function. This approach has been applied in the online re-optimization of a spacecraft attitude controller and numerical results from simulation studies are presented here.

I. INTRODUCTION

MANY difficult real-life control design problems can be formulated in the framework of optimal control theory. Dynamic programming formulation offers the most comprehensive solution to compute nonlinear optimal control in a state feedback form [1]. However, solving the associated Hamilton-Jacobi-Bellman (HJB) equation demands large amounts of computation and storage space dedicated for this purpose. An innovative idea was proposed in [2] to get around this numerical complexity by using an 'Approximate Dynamic Programming (ADP)' formulation. In one version of this approach, called the Dual Heuristic Programming, two neural networks are used to solve for the optimal control solution (Adaptive Critic (AC) design). Optimal solution is reached after the two networks iteratively train each other successfully. There are various types of AC

designs available in literature. An interested reader can refer to [3] for more details.

A significant improvement to the adaptive critic architecture was proposed in [4]. It is named Single Network Adaptive Critic (SNAC) because it uses only the critic network for optimal control solution. SNAC is applicable to control-affine systems for which controllers are synthesized with a quadratic cost function. SNAC eliminates the iterative training loops between the action and critic networks and eliminates the approximation error due to action networks.

There has been a lot of interest in the use of neural networks for direct closed loop controller design that guarantee desired performance in presence of uncertainties and unmodeled dynamics. [5], [6]. An adaptive optimal controller that makes use of online neural networks to approximate parametric/unmodeled nonlinear uncertainties for general control affine systems of the form $X_{k+1} = f(X_k) + g(X_k)U_k$ is developed in this work. The uniqueness of the method proposed in this work is that the online function approximating network can be used to re-optimize in real time an existing Single Network Adaptive Critic [4] based optimal controller that has already been designed for a nominal system. This method is also unique in that unmatched uncertainties can be dealt with. Section 2 discusses approximate dynamic programming and the Single Network Adaptive Critic technique for optimal control design. Section 3 details the online approximation of system uncertainties and the Lyapunov based online weight update rule used in this work. Online re-optimization of the SNAC controller is discussed in section 4. A spacecraft attitude control problem has been chosen as a test bed for the online re-optimization technique. The problem has been presented in section 5. The presence of parametric uncertainties in the system model causes unmodeled nonlinearities to be present in the system equations. The objective is dynamic re-optimization of the SNAC controller designed for a nominal model to make the spacecraft attitude track specific reference values. Results are illustrated in section 6 and conclusions are drawn in section 7.

II. APPROXIMATE DYNAMIC PROGRAMMING

A. Outline

In this section, we attempt to outline the principles of approximate (discrete) dynamic programming, on which the SNAC approach is based on. An interested reader can find

Nishant Unnikrishnan is a Ph. D. candidate at the Mechanical and Aerospace Engineering Department, University of Missouri, Rolla, MO 65401, USA. (e-mail: nu7v3@umr.edu).

S. N. Balakrishnan is a Professor in Aerospace Engineering at the University of Missouri, Rolla, MO 65409, USA. (Tel: 1(573)341-4675, Fax: 1(573)341-4607, E-mail: bala@umr.edu).

Radhakant Padhi is an Assistant Professor in Aerospace Engineering at the Indian Institute of Science, Bangalore, India. (E-mail: padhi@aero.iisc.ernet.in)

more details about the derivations in [7]. In discrete-time formulation, the aim is to find an admissible control U_k , which causes the system described by the *state equation*

$$X_{k+1} = F_k(X_k, U_k) \quad (1)$$

to follow an admissible trajectory that optimizes a sensible performance index J_c given by

$$J_c = \sum_{k=1}^{N-1} \Psi_k(X_k, U_k) \quad (2)$$

where, the subscript k denotes the time step. X_k and U_k represent the $n \times 1$ state vector and $m \times 1$ control vector, respectively, at time step k . The aim is to find U_k as a function of X_k , so that the control can be implemented in the closed loop sense. First, the cost function from time step k is denoted as

$$J_{\alpha_k} = \sum_{k=k}^{N-1} \Psi_k(X_k, U_k) \quad (3)$$

Then J_{α_k} can be rewritten as

$$J_{\alpha_k} = \Psi_k + J_{\alpha_{k+1}} \quad (4)$$

where Ψ_k and $J_{\alpha_{k+1}} = \sum_{k=k+1}^{N-1} \Psi_k$ represent the *utility function* at time step k and the *cost-to-go* from time step $k+1$ to N , respectively. The $n \times 1$ *costate* vector at time step k is defined as

$$\lambda_k = \frac{\partial J_{\alpha_k}}{\partial X_k} \quad (5)$$

For *optimal control (stationary) equation*, the necessary condition for optimality is given by

$$\frac{\partial J_{\alpha_k}}{\partial U_k} = 0 \quad (6)$$

The *optimal control equation* can be written as

$$\left(\frac{\partial \Psi_k}{\partial U_k} \right) + \left(\frac{\partial X_{k+1}}{\partial U_k} \right)^T \lambda_{k+1} = 0 \quad (7)$$

Using Eq.(7), on the *optimal path*, the costate equation can be expressed as

$$\lambda_k = \left(\frac{\partial \Psi_k}{\partial X_k} \right) + \left(\frac{\partial X_{k+1}}{\partial X_k} \right)^T \lambda_{k+1} \quad (8)$$

Eqs.(1), (7) and (8) have to be solved simultaneously, along with appropriate boundary conditions, for the synthesis of optimal control.

B. Single Network Adaptive Critic(SNAC)

The SNAC technique retains all powerful features of the dual network Adaptive Critic (AC) methodology, while eliminating the action network completely. Details of the AC methodology have been provided in [7]. Note that in SNAC design, the critic network captures the functional relationship between X_k and λ_{k+1} , whereas in AC design the critic network capture the relationship between X_k and λ_k . Note that the SNAC method is valid only for the class of problems where the optimal control equation Eq.(7) is explicitly solvable for control variable U_k in terms of the state variable

X_k and costate variable λ_{k+1} . Details regarding the neural network training and convergence checks can be obtained from [4].

III. NEURAL NETWORK BASED ADAPTATION AND ONLINE WEIGHT UPDATE RULE

In this section, we discuss a novel technique that is used to capture parametric uncertainties/ unmodeled nonlinearities that may be present in the plant dynamics but are not considered in the system model used for controller design. The uncertainty approximation is achieved using an online neural network in each system equation.

Consider the class of nonlinear systems with the following structure

$$X_{k+1} = f(X_k) + g(X_k)U_k \quad (9)$$

where $X_k \in R^n$ and $U_k \in R^m$ are the state and control vectors at time step k respectively. The control vector drives the system from an initial point to a final desired point optimizing a sensible performance index J_c given by

$$J_c = \sum_{k=1}^{\infty} \Psi_k(X_k, U_k) \quad (10)$$

It is assumed that a pre-designed SNAC optimal control trajectory U is available to drive the nominal system in Eq.(9) along a desired trajectory. Let the actual plant have the structure

$$X_{k+1} = f(X_k) + g(X_k)U_k + D(X_k) \quad (11)$$

where the controller U_k will have to be re-optimized to optimize the plant performance with the unmodeled dynamics $D(X_k)$ present. Since the term $D(X_k)$ in the plant equation is unknown, the first step in controller re-optimization is to approximate the uncertainty in the plant equation. For this purpose a virtual plant is defined first. Let X^a represent the vector of states of the virtual plant. The dynamics of this virtual plant is governed by

$$X_{k+1}^a = f(X_k) + g(X_k)U_k + \hat{D}(X_k) + K_r(X_k - X_k^a), \quad X^a(0) = X(0) \quad (12)$$

where K_r is a hurwitz matrix. We assume that we have all the actual plant states, X_k , available for measurement at every step. The term $\hat{D}(X_k)$ is the neural network approximation of the unmodeled dynamics of the system which is a function of the actual plant state. Subtracting Eq.(11) from Eq. (12) we have $X_{ak+1} - X_k = \hat{D}(X_k) - D(X_k) + K_r(X_k - X_{ak})$ or $E_{k+1} = \hat{D}(X_k) - D(X_k) - K_r E_k$ where $E_k \equiv X_k^a - X_k$. It can be seen that as $\hat{D}(X_k) - D(X_k)$ approaches zero, the expression becomes an exponentially stable differential equation, *i.e.* $E \rightarrow 0$ as time $k \rightarrow \infty$. Defining $D(X_k) \equiv [d_1(X_k) \dots d_n(X_k)]^T$, where $d_i(X_k)$ denotes the unmodeled dynamics in the differential equation for the i^{th} state of the system. The approach in this study is to have 'n' neural networks (one

for each component of the unmodeled dynamics) for simpler development and analysis.

If all channels are separated, the state equations become

$$x_{k+1} = f(x_k) + g(x_k)u_k + d(X_k) \quad (13)$$

and

$$x_{k+1}^a = f(x_k) + g(x_k)u_k + \hat{d}(X_k) - k_r e_k \quad (14)$$

where $e_k \equiv x_k^a - x_k$.

Subtracting Eq.(13) from Eq.(14) gives

$$x_{k+1}^a - x_k = \hat{d}(X_k) - d(X_k) - k_r e_k \quad (15)$$

Let us assume that there exists a neural network with an optimum set of weights that approximates $d(X_k)$ within a certain accuracy of ε . Thus we have

$$d(X_k) = W^T \varphi(X_k) + \varepsilon \quad (16)$$

Also $\hat{d}(X_k) = \hat{W}_k^T \varphi(X_k)$, where $\hat{W}_k^T \varphi(X_k)$ is the output of the actual neural network. \hat{W}_k represents the actual network weights. Substituting Eq.(16) into Eq.(15) we get

$$x_{k+1}^a - x_{k+1} = \hat{W}_k^T \varphi(X_k) - W^T \varphi(X_k) - \varepsilon_i - k_r e_k \quad (17)$$

or

$$e_{k+1} = \tilde{W}_k^T \varphi(X_k) - \varepsilon_i - k_r e_k \quad (18)$$

where $\tilde{W}_k = \hat{W}_k - W$, is the difference between the optimal weights of the neural network that represents $d(X_k)$ and the actual network weights. Define a series of Lyapunov functions L_i , ($i=1,2,\dots,n$) such that

$$L_k = p e_k^2 + (\tilde{W}_k^T \Gamma^{-1} \tilde{W}_k) \quad (19)$$

where p is a positive definite term and Γ is the learning rate of the neural network.

We can see that

$$L_{k+1} = p e_{k+1}^2 + \tilde{W}_{k+1}^T \Gamma^{-1} \tilde{W}_{k+1} \quad (20)$$

Subtracting Eq. (19) from Eq. (20), we obtain the difference equation

$$\Delta L_k = p e_{k+1}^2 - p e_k^2 + \tilde{W}_{k+1}^T \Gamma^{-1} \tilde{W}_{k+1} - \tilde{W}_k^T \Gamma^{-1} \tilde{W}_k \quad (21)$$

If the weight update rule

$$\tilde{W}_{k+1} = \tilde{W}_k - \Gamma \varphi_k e_{k+1} - \Gamma \sigma \tilde{W}_k \quad (22)$$

is used, it can be shown that the online neural networks approximate the uncertain nonlinearities accurately and all the weights remain bounded. In Eq. (22), σ is a sigma modification constant that is used to ensure weight boundedness even if PE conditions are not fully satisfied. On using Eq. (22) as the weight update rule the difference equation Eq. (21) reduces to

$$\Delta L_k = (p-1+k_r+2\Gamma\alpha)e_{k+1}^2 + (k_r-p)e_k^2 + (\sigma\|W\|^2 + \varepsilon^2) - \sigma(1-2\Gamma\sigma)\|\tilde{W}\|^2 - \sigma\|\tilde{W}\|^2 \quad (23)$$

To ensure a negative ΔL_k , the conditions that need to be satisfied are

$$p+k_r < (1-2\Gamma\alpha), \quad k_r < p, \quad 2\Gamma\sigma < 1 \quad (24)$$

α is an upper bound on $\varphi^T \varphi$. Once the design parameters in the weight update rule are chosen to satisfy the conditions, in Eq.(24), we obtain the inequality condition for ΔL_k as

$$\Delta L_k \leq -M e_k^2 + \beta \quad (25)$$

Eq. (25) can be rewritten as

$$\Delta L_k \leq 0 \text{ if } |e_k| > \sqrt{\beta/M} \quad (26)$$

where

$$M \equiv p - k_r, \quad \beta \equiv \sigma\|W\|^2 + \varepsilon^2 \quad (27)$$

IV. DYNAMIC RE-OPTIMIZATION OF THE SNAC CONTROLLER

In this section we discuss how the costate equations get updated online and how the updating helps in re-optimizing the critic network. The steps of this process are detailed below.

Note from subsection II.B that the main components of the SNAC controller design architecture are the critic network, the optimal control equations, the state equations and the costate equations. Let the plant equation used in the SNAC controller design be given by $X_{k+1} = f(X_k) + g(X_k)U_k$ (Eq. (9)). The performance index used in the SNAC synthesis is

$J_c = \sum_{k=1}^{\infty} \Psi_k(X_k, U_k)$ (Eq. (10)). Let the actual plant equation be written as $X_{k+1} = f(X_k) + g(X_k)U_k + D(X_k)$ (Eq. (11)).

In the SNAC architecture, the critic network is trained to represent the mapping between X_k and λ_{k+1} for the cost function given by Eq. (10) subject to the nominal state equation Eq. (9). The actual plant is given by Eq. (11) where the uncertainty $D(X_k)$ is present in the system dynamics.

The critic network has not been trained with the actual state equation and hence is not the optimal critic for the actual plant. On close examination it can be seen that the costate equations will have to be modified so that an online training routine can help the critic capture the optimal relation between X_k and λ_{k+1} . The uncertainty in the actual plant dynamics is captured by the online neural network and is represented by $\hat{D}(X_k)$. It should be noted here that the inputs to the neural network are the states of the actual plant which we assume are readily available for measurement at every time step.

Revisiting the costate equation (Eq. (8)), it can be seen that there is a term that involves $(\partial \Psi_{k+1} / \partial X_k)$. An essential part of the actual plant equation is the uncertainty $D(X_k)$. This term will have to be incorporated into the costate equation to ensure optimality of the costate. On replacing $D(X_k)$ with $\hat{D}(X_k)$ in Eq. (30) and using it in the costate equation, the new costate equation can be written as

$$\lambda_k = \left(\frac{\partial \Psi_k}{\partial X_k} \right) + \left[\left(\frac{\partial f(X_k)}{\partial X_k} \right) + \left(\frac{\partial g(X_k)}{\partial X_k} \right) U_k + \left(\frac{\partial \hat{D}(X_k)}{\partial X_k} \right) \right]^T \lambda_{k+1} \quad (28)$$

The uncertainty approximation $\hat{d}(X_k)$ for a state equation is given by $\hat{W}_k^T \varphi(X_k)$ (output of the online neural network).

The partial derivative term $(\partial \hat{d}(X_k) / \partial X_k)$ can be written as

$\hat{W}_k^T (\partial \phi(X_k) / \partial X_k)$. Since the basis functions, $\phi(X_k)$ are chosen by the control designer, the partial derivative of the basis functions can be calculated offline. This ensures that the costate equation gets updated online as the online neural network approximates the uncertainty. The reoptimization scheme is represented in Figure 1.

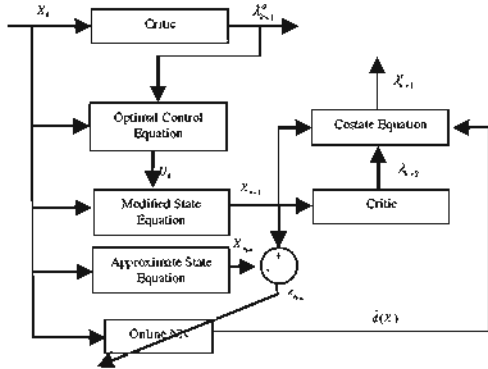


Figure 1: Dynamic Reoptimization of SNAC Controller

The steps for dynamic (online) critic re-optimization are as follows:

1. For each step k , follow the steps below:
 - Input X_k to the critic network to obtain $\lambda_{k+1}^o = \lambda_{k+1}^o$
 - Calculate U_k , form the optimal control equation since X_k and λ_{k+1}^o are known.
 - Get X_{k+1} from the state Eq.(11) using X_k and U_k
 - Get $\hat{d}(X_k)$ as the output of the online neural network
 - Input X_{k+1} to the critic network to get λ_{k+2}^o
 - Using X_{k+1} and λ_{k+2}^o , calculate λ_{k+1}^c from the updated costate Eq.(28)
2. Train the critic network for X_k ; the output being corresponding λ_{k+1}^o .
3. Update time step k to $k+1$
4. Continue steps 1-3.

V. SIMULATION STUDY: SPACECRAFT ATTITUDE CONTROLLER

A. Problem Description and Optimality Conditions

The case study in this paper is a spacecraft attitude controller synthesis [8]. The governing equations of spacecraft dynamics are nonlinear and such control problems cannot be solved with linear control methods.

The rotational motion equation for rigid spacecraft acting under the influence of external torques can be expressed as

$$J \dot{\omega} = -\Omega J \omega + U \quad (29)$$

where J denotes the moment of inertia tensor, ω is the angular velocity vector, and U is the control torque. The skew symmetric matrix Ω is given by

$$\Omega = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \quad (30)$$

The evolution of spacecraft orientation is described in terms of quaternions by the kinematic equations

$$\dot{q} = \frac{1}{2} M(Q) \omega, \quad \dot{q}_4 = -\frac{1}{2} \omega^T q \quad (31)$$

where $q_s = [q^T \ q_4]^T$ is the quaternion and $q = [q_1 \ q_2 \ q_3]^T$ is its vector part. The matrix $M(Q)$ is defined as

$$M(Q) = q_4 I_3 + T \quad (32)$$

where I_3 is the 3x3 identity matrix and the skew symmetric matrix T is given by

$$T = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \quad (33)$$

The four elements of the quaternion are defined as

$$q_i = c_i \sin(\theta/2), \quad q_4 = \cos(\theta/2) \quad i=1,2,3 \quad (34)$$

where θ is the magnitude of the Euler-axis rotation angle and c_i 's are the direction cosines of the Euler axis relative to the reference plane. The target quaternion vector was selected to be $q_t = [0 \ 0 \ 0 \ 1]^T$. The values of the body rates and control torques at equilibrium point are $\omega_e = [0 \ 0 \ 0 \ 0]^T$ and $U_e = [0 \ 0 \ 0]^T$. Next, the deviated state is defined as $X = [\omega_{e1} \ \omega_{e2} \ \omega_{e3} \ q_{e1} \ q_{e2} \ q_{e3} \ q_{e4}]^T$ and deviated control $u \equiv U - U_e$. In terms of these variables, the error dynamics of the system is

$$\begin{aligned} \dot{\omega}_e &= -J^{-1} \Omega J \omega_e + J^{-1} u \\ \dot{q}_e &= 0.5 \begin{bmatrix} q_{e4} \omega_{e1} & -q_{e3} \omega_{e2} & q_{e2} \omega_{e3} \\ q_{e3} \omega_{e1} & q_{e4} \omega_{e2} & -q_{e1} \omega_{e3} \\ -q_{e2} \omega_{e1} & q_{e1} \omega_{e2} & q_{e4} \omega_{e3} \\ -q_{e1} \omega_{e1} & -q_{e2} \omega_{e2} & -q_{e3} \omega_{e3} \end{bmatrix} \end{aligned} \quad (35)$$

where the relation between the error quaternion vector and the spacecraft quaternion vector is

$$q_e = Q_e^{-1} q_s \quad (36)$$

The matrix Q_e is defined as

$$Q_e = \begin{bmatrix} -q_{13} & q_{12} & q_{11} & q_{14} \\ q_{14} & -q_{11} & q_{12} & q_{13} \\ q_{11} & q_{14} & q_{13} & -q_{12} \\ -q_{12} & -q_{13} & q_{14} & -q_{11} \end{bmatrix} \quad (37)$$

Now an optimal regulator problem can be formulated to drive $X \rightarrow 0$ with a cost function, J_c as

$$J_c = \frac{1}{2} \int_0^{\infty} (X^T Q_e X + R_u u^2) dt \quad (38)$$

where $Q_e \geq 0$ and $R_u > 0$ are weighting matrices for state and control respectively. The state equation and cost function were discretized as follows

$$[X_{k+1}] = X_k + \Delta t [f(X_k) + g(X_k) u_k] \quad (39)$$

$$J_c = \sum_{k=1}^{N-1} \frac{1}{2} (X_k^T Q_w X_k + R_w u_k^2) \Delta t \quad (40)$$

Next, using $\Psi_k = (X_k^T Q_w X_k + R_w u_k^2) \Delta t / 2$ in Eqs.(7) and (8), the optimal control and costate equation can be obtained as follows:

$$u_k = -R_w^{-1} (J^{-1})^T \lambda_{k+1} \quad (41)$$

$$\lambda_k = \Delta t Q_w X_k + \left[\frac{\partial F_k}{\partial X_k} \right]^T \lambda_{k+1} \quad (42)$$

In Eq. (42), F_k represents the expression on the right hand side of Eq. (39). For this problem we chose $\Delta t = 0.01$, $Q_w = \text{diag}[10^5 \ 1 \ 1 \ 10^5 \ 10^5 \ 10^5 \ 0]$ and $R_w = 2000 \times I_3$. For SNAC synthesis, we chose seven sub-networks each having a 4-1 structure for the critic network.

B. Uncertainty

The moment of inertia used to design the SNAC controller was

$$J = \begin{bmatrix} 0.49 & 0.02 & -0.03 \\ 0.02 & 0.48 & 0.027 \\ -0.03 & 0.027 & 0.45 \end{bmatrix} \text{kgm}^2 \quad (43)$$

The uncertainties were chosen as

$$\Delta J = \begin{bmatrix} 0.49 \times 0.5 & 0.02 \times 0.3 & -0.03 \times 0.3 \\ 0.02 \times 0.3 & 0.48 \times 0.5 & 0.027 \times 0.3 \\ -0.03 \times 0.3 & 0.027 \times 0.3 & 0.45 \times 0.5 \end{bmatrix} \text{kgm}^2 \quad (44)$$

The aim of the reoptimization scheme was to reoptimize the existing optimal SNAC controller (for the nominal model) to make the spacecraft optimally track the desired reference position. Let $J_n = J + \Delta J$. Additional unknown external torques ($D_i = 0.5 \times [\sin(t) \ \sin(t) \ \sin(t)]^T$) were also introduced in the system body rate equations. Now the new body rate error equation becomes

$$\dot{\omega}_e = -J_n^{-1} \Omega J \omega_e + J_n^{-1} u + D_i \quad (45)$$

Eq. (45) can be expressed in terms of the known model with the uncertainty lumped up in each state equation as shown in Eq. (46).

$$\dot{\omega}_e = -J^{-1} \Omega J \omega_e + J^{-1} u + D \quad (46)$$

It can be seen from Eq. (46) that the unknown terms are essential to obtaining equilibrium conditions. In a discrete format, the structure of the virtual plant used in this problem was

$$\omega_{e,k+1} = \omega_{e,k} + \Delta t (-J^{-1} \Omega J \omega_{e,k} + J^{-1} u_k + \hat{D}_k + K(\omega_{e,k} - \omega_{e,k}^d)) \quad (47)$$

In Eq. (47), $\hat{D} = [\hat{d}_1 \ \hat{d}_2 \ \hat{d}_3]$. \hat{d}_i represents the output of the i^{th} online neural network. Trigonometric basis function neural networks were used in this study for approximating the unmodeled dynamics. Vectors C_i , $i=1,2,3$ which have a structure $C_i = [1 \ \sin(x_i) \ \cos(x_i)]^T$ were created. By using kronecker products [9] to represent the neuron interactions, basis vector Φ was composed as $\Phi = \text{kron}(\text{kron}(C_1, C_2), C_3)$.

The discretized error equations can be written as

$$[X_{k+1}] = X_k + \Delta t [f(X_k) + g(X_k)u_k + [D^T \ 0 \ 0 \ 0]^T] \quad (48)$$

where $[D^T \ 0 \ 0 \ 0]^T$ represents the uncertainty that appears in the error equations. $[\hat{D}^T \ 0 \ 0 \ 0]^T$ represents the vector of neural network outputs. These outputs are used to replace the uncertainties denoted by $[D^T \ 0 \ 0 \ 0]^T$ in Eq. (48), giving rise to

$$[X_{k+1}] = X_k + \Delta t [f(X_k) + g(X_k)u_k + [\hat{D}^T \ 0 \ 0 \ 0]^T] \quad (49)$$

Expression for optimal control is the same as Eqs. (41). The costate equation though changes to

$$\lambda_k = \Delta t Q_w X_k + \left[\frac{\partial F_k}{\partial X_k} \right]^T \lambda_{k+1} \quad (50)$$

where F_k represents the expression on the right hand side of Eq. (49). During each iteration of the simulation, the critic network was updated. The online training was carried out using the error vector (X_k) at that instant as the input and the new target costate (λ_{k+1}) as the output.

VI. RESULTS

Figure 2 illustrates the performance of the three online networks used to approximate the unknown nonlinearities. It is clear that the online networks capture the uncertainties quickly. The states of the spacecraft have been plotted in Figures 3 to 5. Each plot has the nominal state trajectory (state trajectory of the plant if uncertainties were not accounted for) and the state trajectory obtained by online reoptimization of the critic network. Figure 6 details nominal control and reoptimized control trajectories.

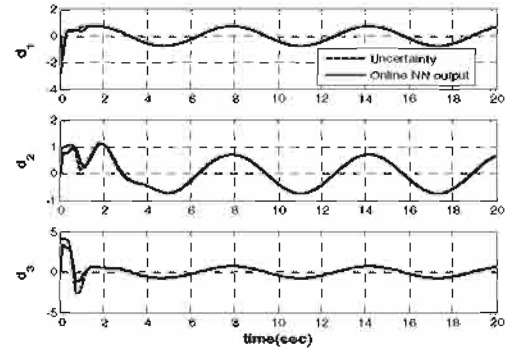


Figure 2: Uncertainty approximation

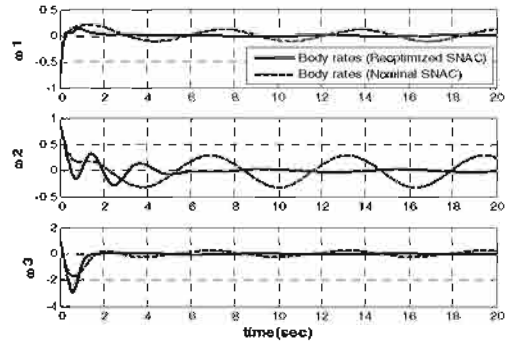


Figure 3: Body rate trajectories

ACKNOWLEDGMENT

This research was supported by NSF grant ECS- 0324428.

REFERENCES

- [1] Bryson, A. E. and Ho, Y. C., "Applied Optimal Control", *Taylor and Francis*, 1975.
- [2] Werbos, P. J., "Approximate Dynamic Programming for Real-time Control and Neural Modeling". In White D.A., & Sofge D.A (Eds.), *Handbook of Intelligent Control, Multiscience Press*, 1992.
- [3] Prokhorov, D. V., and Wunsch, D. C. II, "Adaptive Critic Designs", *IEEE Transactions on Neural Networks*, Vol. 8, 1997, pp. 997-1007.
- [4] Padhi, R., "Optimal Control of Distributed Parameter Systems Using Adaptive Critic Neural Networks", 2001, *Ph.D. Dissertation*, University of Missouri Rolla.
- [5] Lewis, F.L., Jagannathan, S. and Yesildirik, A., "Neural Network Control of Robot Manipulators and Nonlinear Systems", *Taylor and Francis*, UK, 1999.
- [6] Leitner, J., Calise, A. and Prasad, J. V. R., "Analysis of Adaptive Neural Networks for Helicopter Flight Controls", *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, sep-oct 1997, pp. 972-979.
- [7] Balakrishnan, S. N., and Biega, V., "Adaptive-Critic Based Neural Networks for Aircraft Optimal Control", *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 4, July- Aug. 1996, pp. 893-898.
- [8] McDuffie, J. H., and Shtessel, Y. B., "A Decoupled Sliding Mode Controller and Observer for Satellite Attitude Control", *IEEE*, 1997.
- [9] Ham, F. M. and Kostanic, I., "Principles of Neurocomputing for Science and Engineering", *Mc Graw Hill, Inc*, 2001
- [10] Ge, S. S., Lee, T. H., Li, G. Y., and Zhang, J., "Adaptive NN Control for a Class of Discrete-time Nonlinear systems", *Intl. Journ. Of Control*, Vol. 76, No. 4, 2003, pp. 334-354

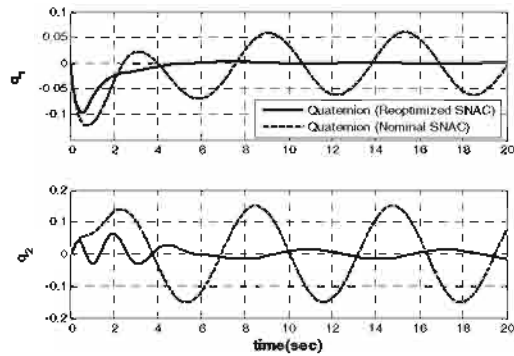


Figure 4: Quaternion trajectories (q_1, q_2)

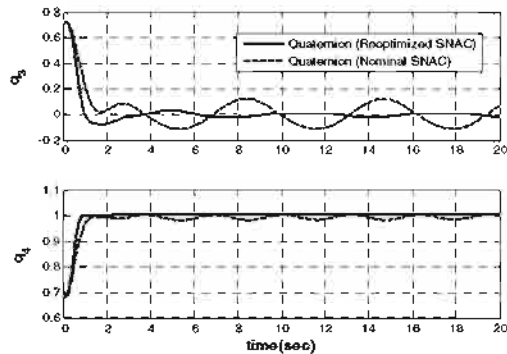


Figure 5: Quaternion trajectories (q_3, q_4)

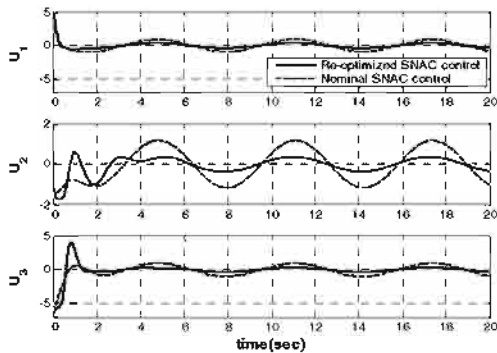


Figure 6: Control trajectories (q_3, q_4)

VII. CONCLUSIONS

Although there have been strategies to compensate for plant uncertainties and to design controllers that stabilize systems, there has not been any concerted effort to dynamically re-optimize the controller in the presence of uncertainties. In this study, we have developed a scheme to re-optimize a pre-designed optimal SNAC controller for control affine systems in the presence of unmodeled/parametric uncertainties. This methodology has been simulated and results have been shown for a spacecraft attitude control problem. This method is unique in that unmatched uncertainties and nonlinearities can be compensated for. The impact of reoptimizing controllers in space operations is critical as it saves a lot of effort (physical and monetary).