Abstract

Dynamic Neural Unit (DNU) network based sub-optimal autopilot design for a tactical missile is investigated. An autopilot is expressed as two-loop nonlinear control phenomena and delivers an approximate solution to value function of State Dependent Riccati Equation (SDRE). The autopilot, not only tracks, stabilizes all the state vectors simultaneously but also, compensates the austere aerodynamic coupling and nonlinearities with an angle-of-attack. The body rates are transformed by outer-loop autopilot using state vectors command (total-angle-of-attack, bank angle), and fin deflections are transformed by inner-loop autopilot using state vectors command (body rates). Reduced cost computation is the novelty of the proposed autopilot. Outcomes are demonstrated using 6-DOF simulations and performance compared with sub-optimal Theta-D autopilot.

Keywords: Autopilot, Sub-optimal, DNU, SDRE, 6-DOF, SDC

1. Introduction

In general, autopilots for tactical missiles are designed based on the assumption that cross-couplings among three axes are negligible. However, there may be large aerodynamic couplings among these axes due to the bank angle which varies with the target maneuver. Among strategies used so far to reduce these coupling effects, the following two ones seem to be dominant. The one is to limit, to some extent, the total-angle-of-attack because the aerodynamic couplings and nonlinearities increase as the angle-of-attack increase, and the other is to select smaller bandwidths of pitch and yaw loops than that of the roll loop. However, these methods are not optimal solutions because the former limits the maneuverability and the latter can make the response of the pitch or yaw loop sluggish.\(^1\)\(^2\)

In this paper, a sub-optimal solution is proposed using DNU networks. As, we know that optimal control methodology needs online computation, and it is essential to solve the algebraic Riccati equation at each sample time. However, it is tough to find the outcome of the SDRE when the dynamics of system become more nonlinear and cross-coupling. Hence, a novelty method for an autopilot is obviously required which is quicker in online computation. The DNU set-up and it's an extensive property as a function approximator, and various control design of systems with functional uncertainty are claimed.\(^3\) As, an adaptivity in real time is the leading apprehension of a DNU set-up autopilot, and with proper training, finds relatively faster and computationally low in cost solution. This solves the value function of SDRE and its performance depends on their weights, somatic, synaptic gains and training approach. An effective second order DNU set-up solves the value function of SDRE.

The paper is structured as follows. Section 2, briefly familiarizes about autopilot design methodology, Section 3, a nonlinear missile dynamics and Section 4 gives the structure of two-loop autopilot with a DNU set-up. Section 5, provides the results and discussion to validate the autopilot design. Finally, conclusions are drawn in Section 6.

*Author for correspondence
2. Sub-optimal Dynamic Neural Unit Autopilot

A single DNU network having two interruption components, three feed-forward components, and two feedback components (make it a second order system) integrated with a plant are shown. When the DNU set-up is not educated, it does not epitomize the value function of SDRE solution i.e. \( u_{m}(l) \neq u_{c}(l) \). A back propagation method used to train the DNU network based on error

\[
\tilde{E}(l) = u_{c}(l) - u_{m}(l);
\]

where \( u_{m}(l) \) is the educated output of DNU. As, we know, if the DNU set-up consequences the reverse of the plant of SDRE, then \( \tilde{E}(l) = u_{c}(l) - u_{m}(l) = 0 \); otherwise \( \tilde{E}(l) = u_{c}(l) - u_{m}(l) \neq 0 \).

The DNU signifies improved solution of the SDRE, as the training continues in superior way. Hence, if endures this progression, the final output of sub-optimal autopilot will be achieved. The initial of \( u_{c} \) is selected very small arbitrary value for the edification of DNU set-up.

The sub-optimal, locally asymptotically stabilizing solution of the nonlinear dynamic system is expressed

\[
u = -R^{-1}g(x)p(x) = -R^{-1}g\frac{\partial v}{\partial x};
\]

the solution of the control law exists when \( P(x) \geq 0 \) in the equation and solution of SDRE of the form \( F^TP + PF - P\Sigma R^{-1}g^TP + Q = 0 \). Though, the criterion of choosing of \( F(x) \) and \( g(x) \) is not distinctive but to pick those must be controllable or at least stabilizable.

2.1. Dynamic Neural Unit for Value Function

The value function for SDRE can be solved using single DNU set-up. As we know \( P(x) \geq 0 \); and,

assuming \( P(x) = \frac{\partial v^*}{\partial x} = \nabla u_{ed} \). We approximate the value function using DNU neural networks. The model of DNU neural set-up can be articulated as

\[
x_{i}(l) = x_{op}(l) + \omega_{op}(l) + x_{ci}(l); \quad \text{where, } l = 0,1,2...k;
\]

\[
v(l) = p_{e}x(l) + p_{x}(l-1) + p_{x}(l-2) - s_{l}v(l-1) - s_{l}v(l-2);
\]

\[
u_{i}(l) = \Psi_{g_{v}}(l); \quad \text{where } x_{i}(l) \in \mathbb{R}^{k} \text{ is input to the DNU, representing the angles (total angle of attack, bank angle), body rate along the body axes and command input. } v(l) \in \mathbb{R}^{l} \text{ is the activity of the neuron, and } u_{i}(l) \in \mathbb{R}^{l} \text{ is the output of DNU set-up.}
\]

3. Missile Dynamics

A mathematical model and assumptions in the derivation for the motion of the missile are considered. From the standard assumptions, the force and moment equations are converted into their aerodynamic coefficients as force and moment coefficients. These are functions of bank angle \( \phi \), total angle of attack \( \alpha \), Mach number \( n \), and control fin deflections \( \delta \), in the wind frame of missile, and are abridged.

\[
\alpha = q_{u} + \frac{Q_{v}}{m_{l}}(-C_{i} \sin \alpha + C_{c} \cos \alpha), \quad \phi = p_{c} + \frac{\partial Q_{\phi}}{\partial \alpha}, \quad \delta = \frac{\partial Q_{\phi}}{\partial \alpha}.
\]

The deflection angles in total-angle-of-attack axis system can be written as follows:

\[
\delta_{t} = \delta_{w} \times \sin \phi, \quad \delta_{\alpha} = \delta_{v} \times \cos \phi, \quad \delta_{w} = \delta_{w} \times \sin \phi, \quad \delta_{v} = \delta_{v} \times \cos \phi, \quad \delta_{w} \times \sin \phi, \quad \delta_{v} \times \cos \phi.
\]

The Angular velocities in body frame system \( p_{b} \times q_{b} \times r_{b} \) are measured around \( x_{b}, y_{b}, z_{b} \) frame are known. Hence, the angular velocities in total-angle-of-attack axis system can be written as follows:

\[
q_{b} = \sin \phi \times \cos \phi - r_{b} \times \sin \phi, \quad r_{b} = \sin \phi \times \sin \phi + r_{b} \times \cos \phi, \quad p_{b} = \sin \phi.
\]

Since, aerodynamic coefficients in the total-angle-of-attack axis system are \( C_{r}, C_{m}, C_{n}, C_{y}, C_{x}, C_{z} \). These coefficients are obtained through a polynomial fit in terms of above said dependencies. The expressions for force coefficients (axial, normal and side) and moment coefficients (pitching, yawing and rolling) are set as:

\[
C_{x} = f(C_{x_1, n}, \delta_{c}, \delta_{t}, \delta_{m}, \delta_{p}, \alpha), \quad C_{y} = f(C_{y_1, n}, \delta_{c}, \delta_{t}, \delta_{m}, \alpha), \quad C_{z} = f(C_{z_1, n}, \delta_{c}, \delta_{t}, \delta_{m}, \alpha);
\]

\[
C_{m} = f(C_{m_1, n}, \delta_{c}, \delta_{t}, \delta_{m}, \alpha).
\]

These coefficients are then converted into aerodynamic and moment coefficients \( C_{x}, C_{y}, C_{z}, C_{m}, C_{n} \) in the body
frame system. The conversion between the two axis systems is as follows: 
\[ C_x = C_y = C_z = \cos\phi_a + C_z \sin\phi_a; \]
\[ C_y = -C_y \times \sin\phi_a + C_z \cos\phi_a \text{ and } C_y = C_y; \]
\[ C_z = -C_z \times \sin\phi_a + C_z \cos\phi_a; \] 
finally, aerodynamic forces and moments along the \( x_n, y_n, z_n \) axes are deliberated.

4. Two-loop Autopilot

The design of two-loop autopilot is revealed in Figure 1. Firstly, a guidance command (acceleration) from the guidance laws of missile are converted to bank angle (\( \phi_c \)), total angle-of-attack (\( \alpha_c \)), commands using BTT/STT logic. The outer-loop autopilot converts \( \phi_c, \alpha_c \) commands to body rates (roll-rate \( (p_c) \), pitch-rate \( (q_c) \), and yaw-rate \( (r_c) \)) commands for the inner-loop. The inner loop autopilot then converts \( p_c, q_c, r_c \) to canard \( \delta_c, \delta_y \) and tail \( \delta_t, \delta_r \) commands accordingly for actuators.

4.1. Outer-loop

For the outer-loop, its state, control vector and tracking command are chosen as: \( \bar{x} = [\phi, \alpha, \phi, \phi] \), \( u = [p_{ol}, q_{ol}, r_{ol}] \) and \( r_c = [\phi_c, \alpha_c] \) respectively. The outer-loop controller becomes:
\[ u^*_{ol} = -\frac{1}{2} \bar{R}_{ol}(\bar{x})^T \bar{B}_{ol}(\bar{x}) \nabla u^*_{ol, DNU} \begin{bmatrix} x_r - r_c \ x_N \end{bmatrix}; \] 
where
\[ \nabla u^*_{ol, DNU} = \Psi [g_{\phi_c, \alpha_c} (k)]; x_r = [\phi, \alpha, \phi, \phi]^T, x_N = [\alpha, \phi, \phi] \] ; As, we know from the state dependent coefficient (SDC) parameterization techniques, there are several traditions of factoring \( f(x) \) as \( A(x)x \). Finally, the outer-loop SDC factorization for \( \bar{A}_{ol}(\bar{x}) \) and \( \bar{B}_{ol}(\bar{x}) \) are designed as given below:

\[ \bar{A}_{ol}(\bar{x}) = \begin{bmatrix} a_{ol,11} & a_{ol,12} & 0 & 0 \\ a_{ol,21} & a_{ol,22} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; \]
\[ \bar{B}_{ol}(\bar{x}) = \begin{bmatrix} 1 & 0 & -\cot(\alpha) \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

4.2. Inner-loop

The second loop of autopilot is called inner-loop. This loop has to track the commands of body rates (roll rate \( p_c \), pitch rate \( q_c \) and yaw rate \( r_c \)), which are achieved from the first-loop autopilot. Let, the state, control vector and tracking command are given for inner-loop autopilot by \( [r, s] \) \( : u_c = [\delta_c, \delta_y, \delta_t, \delta_r]^T; r_c = [p_c, q_c, r_c]^T \). Hence, the autopilot of inner-loop is
\[ u^*_{il} = -\frac{1}{2} \bar{R}_{il}(\bar{x})^T \bar{B}_{il}(\bar{x}) \nabla u^*_{il, DNU} \begin{bmatrix} x_r - r_c \ x_N \end{bmatrix}; \] 
where
\[ \nabla u^*_{il, DNU} = \Psi [g_{\delta_c, \delta_y, \delta_t, \delta_r} (k)]; x_r = [p, q, r]^T, x_N = [s]^T \] ; same factorization technique used to get corresponding inner-loop SDC factorization;

\[ \bar{A}_{il}(\bar{x}) = \begin{bmatrix} c_{il,11} & 0 & 0 & c_{il,14} \\ c_{il,21} & c_{il,22} & c_{il,23} & c_{il,24} \\ c_{il,31} & c_{il,32} & c_{il,33} & c_{il,34} \\ 0 & 0 & 0 & -\lambda \end{bmatrix}; \]
\[ \bar{B}_{il}(\bar{x}) = \begin{bmatrix} d_{il,21} & d_{il,22} & d_{il,23} & d_{il,24} & 0 \\ d_{il,31} & d_{il,32} & d_{il,33} & d_{il,34} & d_{il,35} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]

5. Results and Discussion

The two-loop structure of autopilot with plant (6-DOF) is shown in Figure 1. In order to show the performance, simulations under following conditions are performed: Considering a tactical missile (canard and tail fins) at Mach number \( n=2.7 \) and height \( h=20,000 \) ft.

At the start of simulation, a guidance command of acceleration \( (a_c = 10g, a = -10g) \) is commanded into the outer loop of autopilot. A maximum hard limit 300/\( s \) is obligatory upon demanded body rates \( (p, q \text{ and } r) \). Also maximum fin deflection (canard and tail) limit is set to 45° respectively. The following state, control and DNU value function weighting matrices were selected for the first-loop of autopilot:
Sub-Optimal Autopilot Design based on Dynamic Neural Unit Network for Tactical Missile

\[ \tilde{Q}(\bar{x})_{OL} = \begin{bmatrix} q_{OL11} & 0 & 0 & 0 \\ 0 & q_{OL22} & 0 & 0 \\ 0 & 0 & q_{OL33} & 0 \\ 0 & 0 & 0 & q_{OL44} \end{bmatrix}; \]

\[ \tilde{R}(\bar{x})_{OL} = \begin{bmatrix} r_{OL11} & 0 & 0 \\ 0 & r_{OL22} & 0 \\ 0 & 0 & r_{OL33} \end{bmatrix}; \] where

\[ q_{OL11}(\bar{x}) = 0.32828 \times 27000; \quad r_{OL11}(\bar{x}) = 0.3/8; \]

\[ q_{OL22}(\bar{x}) = 3.6475; \quad r_{OL22}(\bar{x}) = 0.02/2; \quad q_{OL33}(\bar{x}) = 3.6475 \times 10^{3}; \quad r_{OL33}(\bar{x}) = 150; \]

\[ q_{OL44}(\bar{x}) = 0.0684 \times 20; \quad r_{OL44}(\bar{x}) = 0.2; \quad q_{OL55}(\bar{x}) = 0.5; \quad r_{OL55}(\bar{x}) = 1; \]

\[ q_{IL11}(\bar{x}) = 0.0684 \times 4/5; \quad r_{IL11} = 1; \quad q_{IL33} = 0.4; \quad q_{IL55} = 0.0684 \times 4/5; \quad r_{IL33} = 0.2; \quad r_{IL55} = 0.5; \]

The weightings for inner-loop are selected in an analogous way as did in the outer-loop autopilot;

\[ \tilde{Q}(\bar{x})_{IL} = \begin{bmatrix} q_{IL11} & 0 & 0 & 0 \\ 0 & q_{IL22} & 0 & 0 \\ 0 & 0 & q_{IL33} & 0 \\ 0 & 0 & 0 & q_{IL44} \end{bmatrix}; \]

\[ \tilde{R}(\bar{x})_{IL} = \begin{bmatrix} r_{IL11} & 0 & 0 & 0 \\ 0 & r_{IL22} & 0 & 0 \\ 0 & 0 & r_{IL33} & 0 \\ 0 & 0 & 0 & r_{IL44} \end{bmatrix}; \] where

\[ q_{IL11} = 0.0684 \times 4/5; \quad r_{IL11} = 1; \quad q_{IL22} = 0.0684 \times 4 \times 50; \quad r_{IL22} = 1; \]

\[ q_{IL33} = 0.0684 \times 4/5; \quad r_{IL33} = 0.4; \quad q_{IL44} = 0.0684 \times 4/5; \quad r_{IL44} = 0.2; \quad r_{IL55} = 0.5; \]

\[ q_{IL55} = 0.0684 \times 4/5; \quad r_{IL55} = 0.2; \quad p_1 = 0.2; \quad p_2 = 0.2; \quad s_1 = 0.005; \quad s_2 = 0.005 \]

The inner-loop tracking, i.e., \( p \) and \( q \), is investigated. Figure 7-8 shows the required fin deflections. It can be comprehended that the behavior of fin deflections are meeting the design constraints and bounded at this operating load conditions.

From the simulations, several observations are investigated. First, the DNU autopilot of outer and inner-loops
show excellent tracking and stability characteristics. However, stability proof is not done in this paper. Second, the DNU based autopilot design is sub-optimal which is faster than Theta-D controller. The third benefit of the DNU autopilot technique is its online computation that does not need excessive computations. Finally, this autopilot design does not require SDRE solution online which involves four by four matrices in both the loops.

6. Conclusions

In this paper, a DNU set-up based autopilot has been derived for a tactical missile which is sub-optimal. The two-loop autopilot is designed based on SDRE technique and SDRE value function is solved online by using DNU set-up. The simulation results show the effectiveness of its tracking and stability measure. The DNU network gives better an online outcomes to this nonlinear optimal control which is compared with Theta-D suboptimal control.

7. Acknowledgement

The author wish to express his deep sense of gratitude to Director, RCI for providing opportunity to work on this problem.

8. References

1. J.R. Cloutier, D.T. Stansbery. Nonlinear, Hybrid Bank-to-Turn/Skid-to-Turn Autopilot Design. In: Proc. of the AIAA Guidance, Navigation, and Control Conference, Montreal, Canada. 2001.
2. J.R. Cloutier, D.T. Stansbery. The Capabilities and Art of State Dependent Riccati Equation-Based Design. In: Proc. of the American Control Conference, Anchorage, AK, USA. 2002b; 86–91.
3. M. Xin, S.N. Balakrishnan, D.T. Stansbery, E.J. Ohlmeyer. A New Method for Suboptimal Control of A Class of Nonlinear Systems. Proceedings of IEEE Conference on Decision and Control, Las Vegas, Nevada. Dec 2002;10–13.
4. S. Das, M. Sinha, A. Misra. Dynamic Neural Units for Adaptive Magnetic Attitude Control for Spacecraft. Journal of Guidance, Control, and Dynamics.2012; 35(4):1280–1291.
5. C. Song, Y.S. Kim. A New Approach to Motion Modeling and Autopilot Design of Skid-to-Turn. AIAA, Guidance, Navigation and Control Conference, San Diego, CA. July1996; 29–31:3916.