01 Jan 2022

# Neural Network Attitude Control System Design for the Wallops Arc-Second Pointer

Pavel Galchenko

Henry J. Pernicka
*Missouri University of Science and Technology*, pernicka@mst.edu

## Recommended Citation

P. Galchenko and H. J. Pernicka, "Neural Network Attitude Control System Design for the Wallops Arc-Second Pointer," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 7, pp. 1365 - 1370, American Institute of Aeronautics and Astronautics, Jan 2022.
The definitive version is available at https://doi.org/10.2514/1.G006465

Journal of Guidance, Control, and Dynamics
Vol. 45, No. 7, July 2022

# Engineering Notes

## Neural Network Attitude Control System Design for the Wallops Arc-Second Pointer

Pavel Galchenko* and Henry Pernicka†
*Missouri University of Science and Technology, Rolla, Missouri 65409-0050*

https://doi.org/10.2514/1.G006465

### Nomenclature

| | | |
|---|---|---|
| $A_{\mathrm{CL}}$ | = | closed-loop dynamics |
| $B_w$ | = | system bandwidth, Hz |
| $\boldsymbol{e}$ | = | error states; $[\boldsymbol{e}_1 \;\; \boldsymbol{e}_2]^T$ |
| $\boldsymbol{e}_a$ | = | observer estimation error |
| $\mathcal{F}_x, \hat{\boldsymbol{\mathcal{F}}}_x$ | = | true and estimated dynamics |
| $\tilde{\boldsymbol{\mathcal{F}}}_x$ | = | dynamics estimation error |
| $\boldsymbol{f}_d$ | = | desired dynamics |
| $\boldsymbol{f}_x(\boldsymbol{x}_2)$ | = | known system dynamics |
| $\boldsymbol{g}_x$ | = | control mapping to dynamics |
| $H_\beta, H_\gamma$ | = | pitch/yaw angles from gimbal hubs [rad] |
| $J$ | = | system inertia matrix, slug $\cdot$ ft$^2$ |
| $K_p, K_i, K_d$ | = | proportional $(\mathrm{ft}\cdot\mathrm{lb})/\mathrm{rad}$, integral $\mathrm{ft}\cdot\mathrm{lb}/(\mathrm{rad}\cdot\mathrm{s})]$, and derivative $\mathrm{ft}\cdot\mathrm{lb}/(\mathrm{rad/s})$ gains |
| $K_v, K_z$ | = | controller gain matrices |
| $K_2$ | = | linear observer gain matrix |
| $L$ | = | Lyapunov function |
| $M, N$ | = | neural network adaptation gain matrices |
| $\boldsymbol{r}$ | = | filtered tracking error |
| $tr\{\}$ | = | trace operator |
| $\boldsymbol{u}$ | = | roll, pitch, and yaw control torques; $[u_\alpha \;\; u_\beta \;\; u_\gamma]^T$, ft $\cdot$ lb |
| $\boldsymbol{u}_r$ | = | robustifying control |
| $W^T, V^T, \hat{W}^T, \hat{V}^T$ | = | true and estimated neural network weighting matrices |
| $\tilde{W}, \tilde{V}$ | = | neural network weight matrices estimation error |
| $\boldsymbol{x}_d$ | = | desired orientation [rad] and angular rates [rad/s]; $[\boldsymbol{x}_{d1} \;\; \boldsymbol{x}_{d2}]^T$ |
| $\boldsymbol{x}_1$ | = | roll, pitch, and yaw rotations; $[\theta_\alpha \;\; \theta_\beta \;\; \theta_\gamma]^T$, rad |
| $\boldsymbol{x}_2$ | = | roll, pitch, and yaw angular velocities; $[\omega_\alpha \;\; \omega_\beta \;\; \omega_\gamma]^T$, rad/s |
| $\Gamma$ | = | observer neural network adaptation rate |
| $\Delta(x)$ | = | unmodeled perturbations/dynamics |
| $\boldsymbol{\delta}(\boldsymbol{x})$ | = | known disturbance dynamics |
| $\varepsilon$ | = | bounded neural network estimation error |
| $\kappa$ | = | design parameter |
| $\lambda$ | = | gain coefficient |
| $\sigma, \sigma'$ | = | activation function and its derivative |
| $\phi(z)$ | = | basis vector |
| $\omega_n, \zeta, \alpha$ | = | controller natural frequency, damping ratio, and tuning parameter |

## I. Introduction

THE Wallops Arc-Second Pointer (WASP) system provides a platform to point scientific instruments with subarcsecond accuracy and stability while suspended from a high-altitude balloon (HAB) to conduct novel science in the Earth's upper atmosphere [1,2]. The WASP system is built around an external gondola structure, which is suspended from a rotator gimbal attached to the HAB. The rotator provides initial targeting and coarse azimuth tracking and stabilization for the external gondola. The centerbody is attached to an inner gimbal frame and the external gondola via torque motor hubs to provide pitch-yaw articulated axes.

Over the 10-plus years of operations the WASP platform conducted seven science missions, continuously demonstrating performance in the arcsecond to subarcsecond regime [3–6]. While significant improvements in the pointing performance would likely require hardware modifications, it is still desirable to determine if pointing performance could be improved through controller design. This work suggests to augment WASP with the addition of a neural network control system, which can be activated in lieu of the traditional proportional–integral–derivative (PID) system.

This research offers several contributions to the nonlinear control field for gimbaled pointing systems. Two neural network methodologies are introduced, formulated, and tested for a gimbaled control system in a high fidelity environment. Specific contributions include the practical application to a physical gimbaled system, simplification of the dynamics through modification of the parameterization matrix, formulating a vector filtered tracking error, and reformulating the original MSO method to improve robustness as learning gains are increased.

## II. Background

A typical attitude dynamics model [7,8] can be given by a two-vector state system as

$$\dot{\boldsymbol{x}}_1 = B(\theta_\alpha, \theta_\gamma)\boldsymbol{x}_2$$

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{f}_x(\boldsymbol{x}_2) + \boldsymbol{\delta}(\boldsymbol{x}) + \boldsymbol{g}_x\boldsymbol{u},$$

$$\text{where } B(\theta_\alpha, \theta_\gamma) = \begin{bmatrix} 1 & -\cos\theta_\alpha \tan\theta_\gamma & \sin\theta_\alpha \tan\theta_\gamma \\ 0 & \cos\theta_\alpha \sec\theta_\gamma & -\sin\theta_\alpha \sec\theta_\gamma \\ 0 & \sin\theta_\alpha & \cos\theta_\gamma \end{bmatrix} \qquad (1)$$

and where $\boldsymbol{f}_x(\boldsymbol{x}_2) = -J^{-1}[\boldsymbol{x}_2 \times]J\boldsymbol{x}_2$, $\boldsymbol{x}_1 = [\theta_\alpha \;\; \theta_\beta \;\; \theta_\gamma]^T$, and $\boldsymbol{x}_2 = [\omega_\alpha \;\; \omega_\beta \;\; \omega_\gamma]^T$, $\boldsymbol{u} = [u_\alpha \;\; u_\beta \;\; u_\gamma]^T$, where $J$ is the constant inertia matrix of the rigid body, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are the rotation and angular velocity vectors, respectively, of the body with respect to the inertial frame, $\boldsymbol{\delta}(\boldsymbol{x})$ is some known disturbance dynamics, $\boldsymbol{g}_x$ maps the controller dynamics to the system, and $\boldsymbol{u}$ is the control input. It should be noted that $J^{-1}$ is also embedded into the disturbance dynamics $\boldsymbol{\delta}(\boldsymbol{x})$ and the mapping of the controller dynamics, $\boldsymbol{g}_x$, to simplify the formulation.

Though a variety of models exist to describe the dynamics of two-axis gimbal systems [9–12], they can be simplified to the form given in Eq. (1) with variations in the attitude parameterization

Received 23 September 2021; revision received 18 January 2022; accepted for publication 18 January 2022; published online 14 February 2022.

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, 400 West 13th Street.

†Curators' Distinguished Teaching Professor, Department of Mechanical and Aerospace Engineering, 400 West 13th Street.

matrix, $B(\theta_\alpha, \theta_\gamma)$, and definitions of $\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}_2)$ and $\boldsymbol{\delta}(\boldsymbol{x})$. Reductions in cross-coupling terms in the kinematics are reduced through mechanical design, where the centerbody roll angle is fixed and null in the gimbal frame, such that $\theta_\alpha = 0$. The yaw rotation of the centerbody is always small within the gimbal frame due to the rotator, which historically keeps yaw less than 1.0 deg during pointing operations. Using the null roll angle and small angle approximation, the parameterization matrix $B(\theta_\alpha, \theta_\gamma)$ becomes the identity matrix $I_{3\times 3}$ in Eq. (1), such that the simplified dynamics become

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 &= \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}_2) + \boldsymbol{\delta}(\boldsymbol{x}) + \boldsymbol{g}_{\boldsymbol{x}}\boldsymbol{u} \end{aligned} \tag{2}$$

and are in Brunovsky canonical form [13]. Although effort has been taken to present the formulation of the system dynamics, the new methodology introduced in this research makes no assumptions on the details of $\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}_2)$ and $\boldsymbol{\delta}(\boldsymbol{x})$ and only assumes that the dynamics can be described by the form given in Eq. (2).

### A. Error Dynamics and Input–Output Feedback Linearization

The error dynamics are given by

$$\begin{aligned} \dot{\boldsymbol{e}}_1 &= \boldsymbol{e}_2 \\ \dot{\boldsymbol{e}}_2 &= \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}) - \boldsymbol{f}_d + \boldsymbol{\delta}(\boldsymbol{x}) + \boldsymbol{g}_{\boldsymbol{x}}\boldsymbol{u} \end{aligned} \tag{3}$$

where $\boldsymbol{e}_1 = \boldsymbol{x}_1 - \boldsymbol{x}_{d1}$ and $\boldsymbol{e}_2 = \boldsymbol{x}_2 - \boldsymbol{x}_{d2}$, where $\boldsymbol{x}_{d1}$ and $\boldsymbol{x}_{d2}$ are the desired states, and $\boldsymbol{f}_d$ are the desired dynamics. Note that for a regulatory reference system, $\boldsymbol{x}_{d2} = 0$ and $\boldsymbol{f}_d = 0$. By using input–output feedback linearization [14], the control input becomes $\boldsymbol{u} = (1/\boldsymbol{g}_{\boldsymbol{x}})(-\boldsymbol{f}_x(\boldsymbol{x}) + \boldsymbol{f}_d - \boldsymbol{\delta}(\boldsymbol{x}) - \boldsymbol{k}_p\boldsymbol{e}_1 - \boldsymbol{k}_d\boldsymbol{e}_2)$ where $\boldsymbol{k}_p$ and $\boldsymbol{k}_d$ are proportional and derivative controller gain vectors, respectively, and $|\boldsymbol{g}_{\boldsymbol{x}}| > 0$. The closed-loop error dynamics in Eq. (3) are reduced to $\dot{\boldsymbol{e}} = A_{\mathrm{CL}}\boldsymbol{e}$, such that the origin of the unperturbed nonlinear system will be asymptotically stable if $A_{\mathrm{CL}}$ is Hurwitz. This method of nonlinear control is the basis from which the neural network controllers are designed.

### B. Heritage Controller Design

The WASP platform uses a PID controller to achieve subarcsecond pointing accuracy and stability. Using the system plant $P(s) = J_i^{-1}/s^2$, where $J_i$ is the inertia of specified axis, and choosing a PID control input, the closed-loop characteristic equation becomes $s^3 + [J_i^{-1}K_d]s^2 + [J_i^{-1}K_p]s + [J_i^{-1}K_i]$ where $K_p$, $K_i$, and $K_d$ are proportional, integrator, and derivative gains, respectively. Using the typical $(\zeta, \omega_n)$ parameterization and fixing the third pole at $-\alpha\omega_n$, controller gains can be selected by

$$K_p = \frac{(2\zeta\alpha + 1)\omega_n^2}{J_i^{-1}} \quad K_i = \frac{\alpha\omega_n^3}{J_i^{-1}} \quad K_d = \frac{(2\zeta + \alpha)\omega_n}{J_i^{-1}} \tag{4}$$

where $\omega_n$ is the natural frequency of the system, $\zeta$ is the damping ratio, and $\alpha$ is a design parameter. By fixing the design parameter at $\alpha = 1$, the values for $\zeta$ and $\omega_n$ are tuned to give desired performance characteristics.

## III. Neural Network Design

It has been shown that, with the formulation of the dynamics and the input–output feedback linearization controller, the rotational systems error dynamics have guaranteed asymptotic stability if $\boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}_2)$ and $\boldsymbol{\delta}(\boldsymbol{x})$ are perfectly known. Assume that the true dynamics are given as $\boldsymbol{\mathcal{F}}_{\boldsymbol{x}} = \boldsymbol{f}_{\boldsymbol{x}}(\boldsymbol{x}_2) + \boldsymbol{\delta}(\boldsymbol{x}) + \Delta(\boldsymbol{x})$, where $\Delta(\boldsymbol{x})$ is a set of all other unmodeled dynamics. Using neural networks, estimating the true dynamics can be attempted using $\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}}$. A one-layer neural network is given by

$$\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} = \hat{W}^T\phi(z) \quad \text{while } \boldsymbol{\mathcal{F}}_{\boldsymbol{x}} = W^T\phi(z) + \varepsilon \tag{5}$$

and a two-layer neural network is given as

$$\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} = \hat{W}^T\sigma(\hat{V}^T\phi(z)) \quad \text{while } \boldsymbol{\mathcal{F}}_{\boldsymbol{x}} = W^T\sigma(V^T\phi(z)) + \varepsilon \tag{6}$$

where $\sigma$ is some activation function, $\phi(z)$ is the basis vector, $W^T$ and $V^T$ are the true weights that give the value of $\boldsymbol{\mathcal{F}}_{\boldsymbol{x}}$ with some bounded approximation error $\varepsilon$ through the universal function approximation properties of neural networks [13], and $\hat{W}^T$ and $\hat{V}^T$ are the approximated weights.

The two-vector state systems from Eq. (2) can now be rewritten as

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= \boldsymbol{x}_2 \\ \dot{\boldsymbol{x}}_2 &= \boldsymbol{\mathcal{F}}_{\boldsymbol{x}} + \boldsymbol{g}_{\boldsymbol{x}}\boldsymbol{u} \end{aligned} \tag{7}$$

and the error dynamics in Eq. (3) as

$$\begin{aligned} \dot{\boldsymbol{e}}_1 &= \boldsymbol{e}_2 \\ \dot{\boldsymbol{e}}_2 &= \boldsymbol{\mathcal{F}}_{\boldsymbol{x}} - \boldsymbol{f}_d + \boldsymbol{g}_{\boldsymbol{x}}\boldsymbol{u} \end{aligned} \tag{8}$$

forming the foundation of the two neural network methodologies studied in this research. In the first method, learning is driven by the tracking error dynamics, Eq. (8), whereas the second method introduces an observer for the state dynamics, Eq. (7), and learning is driven by the estimation error dynamics.

### A. SNNARC Formulation

The first method, originally developed for robotic manipulators [13], is the Subarcsecond Neural Network Attitude Reference Controller (SNNARC). The error dynamics from Eq. (8) are written as a filtered tracking error where

$$\begin{aligned} \boldsymbol{r} &= \boldsymbol{e}_2 + \lambda\boldsymbol{e}_1 \\ \dot{\boldsymbol{r}} &= \boldsymbol{\mathcal{F}}_{\boldsymbol{x}} - \boldsymbol{f}_d + \lambda\boldsymbol{e}_2 + \boldsymbol{g}_{\boldsymbol{x}}\boldsymbol{u} \end{aligned} \tag{9}$$

where $\lambda$ is some positive gain coefficient and $\boldsymbol{r}$ is now a vector of scalar errors for each control axis. It is now possible to use the estimated dynamics $\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}}$ in the controller design such that

$$\begin{aligned} \boldsymbol{u} &= \frac{1}{\boldsymbol{g}_{\boldsymbol{x}}}(-\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} + \boldsymbol{f}_d - K_v\boldsymbol{r} - \lambda\boldsymbol{e}_2 + \boldsymbol{u}_r) \\ \text{where } \boldsymbol{u}_r &= -K_z(\|\hat{\Theta}\| + \Theta_m)\boldsymbol{r} \end{aligned} \tag{10}$$

where $K_v$ and $K_z$ are some positive controller gain matrices, $|\boldsymbol{g}_{\boldsymbol{x}}| > 0$, and the weighting matrices are defined as

$$\Theta = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \quad \text{and } \hat{\Theta} = \begin{bmatrix} \hat{V} & 0 \\ 0 & \hat{W} \end{bmatrix} \quad \text{and } \|\Theta\| \le \Theta_m$$

where $\Theta_m$ is a bound on the weighting matrices. The closed-loop system becomes

$$\dot{\boldsymbol{r}} = -K_v\boldsymbol{r} + \tilde{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} + \boldsymbol{u}_r \tag{11}$$

where $\tilde{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}}$ is the estimation error defined by $\tilde{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} = \boldsymbol{\mathcal{F}}_{\boldsymbol{x}} - \hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}}$. Using a two-layer neural network, Eq. (6), weight update laws are selected as

$$\begin{aligned} \dot{\hat{W}} &= M(\hat{\sigma} - \hat{\sigma}'\hat{V}^T\phi(z))\boldsymbol{r}^T - \kappa\|\boldsymbol{r}\|M\hat{W} \\ \dot{\hat{V}} &= N\phi(z)\boldsymbol{r}^T\hat{W}^T\hat{\sigma}' - \kappa\|\boldsymbol{r}\|N\hat{V} \end{aligned} \tag{12}$$

where $M$ and $N$ are positive definite matrices, $\hat{\sigma} = \sigma(\hat{V}^T\phi(z))$, $\hat{\sigma}' = \sigma'(\hat{V}^T\phi(z))$, and $\kappa$ is a design parameter such that $\kappa > 0$.

To prove the stability of the system, the candidate Lyapunov function is selected as

$$L = \frac{1}{2}\boldsymbol{r}^T\boldsymbol{r} + \frac{1}{2}tr\{\tilde{W}^T M^{-1}\tilde{W}\} + \frac{1}{2}tr\{\tilde{V}^T N^{-1}\tilde{V}\} \tag{13}$$

where $tr\{\}$ is the trace operator. By taking the time derivative of the Lyapunov function in Eq. (13) and using the definitions of $\boldsymbol{r}$ and $\dot{\boldsymbol{r}}$ in Eqs. (9) and (11), respectively, and the weight update law in Eq. (12), stability in a compact set about the origin can be shown [15]. Though the full proof is omitted for brevity, it follows similar methods in literature [13,16], showing stability in a compact set about the origin if either $\|\boldsymbol{r}\| > \delta_r$ or $\|\tilde{\Theta}\| > \delta_\Theta$, where $\delta_r$ and $\delta_\Theta$ are positive constants. This demonstrates that the Lyapunov function is ultimately upper bounded (UUB) in both $\|\boldsymbol{r}\|$ and $\|\tilde{\Theta}\|$ and thus showing Lyapunov stability of the proposed SNNARC method.

### B. MSO Formulation

The second method, originally developed for estimating unmodeled dynamics [17–19], uses a modified state observer (MSO) and an input–output feedback linearization controller. The kinematics in Eq. (7) are ignored and the system is rewritten as single-vector system as

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{\mathcal{F}_x} + \boldsymbol{g_x u} \tag{14}$$

where $\dot{\boldsymbol{x}}_2$ is identical to the previous formulation. An observer can be designed as

$$\dot{\hat{\boldsymbol{x}}} = \hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} + \boldsymbol{g_x u} + K_2(\boldsymbol{x} - \hat{\boldsymbol{x}}) \tag{15}$$

where $\hat{\boldsymbol{x}}$ is the estimated state and $K_2$ is some gain matrix. In previous literature, a single layer neural network, Eq. (5), has been used to estimate the nonlinear dynamics $\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}}$ for the observer. The state, Eq. (14), and observer, Eq. (15), are rewritten in terms of observer error, $\boldsymbol{e}_a = \boldsymbol{x} - \hat{\boldsymbol{x}}$, such that the observer error dynamics are given as

$$\dot{\boldsymbol{e}}_a = -K_2\boldsymbol{e}_a + \tilde{W}^T\phi(z) + \varepsilon \tag{16}$$

where $\phi(z)$ is the basis vector and $\tilde{W}$ is the error between the true weights and estimated weights such that $\tilde{W} = W - \hat{W}$. The neural network weight update law is given as

$$\dot{\hat{W}} = \Gamma\phi(z)\boldsymbol{e}_a^T - \kappa\|\boldsymbol{e}_a\|\Gamma\hat{W} \tag{17}$$

where $\Gamma$ is the adaptation rate and $\kappa$ is the modification factor, which bound the weights and provides robustness. The stability of the system in Eq. (16) can be shown with the candidate Lyapunov function

$$L = \frac{1}{2}\boldsymbol{e}_a^T\boldsymbol{e}_a + \frac{1}{2}tr\left\{\tilde{W}^T\Gamma^{-1}\tilde{W}\right\} \tag{18}$$

where $tr\{\}$ is the trace operator. By taking the time derivative of the Lyapunov function in Eq. (18) and using the definitions of $\boldsymbol{e}_a$ and $\dot{\boldsymbol{e}}_a$ in Eq. (16), and the weight update law in Eq. (17), stability in a compact set about the origin can be shown [15]. Though the full proof is omitted for brevity, it follows similar methods in literature [20], showing stability in a compact set about the origin if either $\|\boldsymbol{e}_a\| > \delta_{ea}$ or $\|\tilde{W}\| > \delta_W$, where $\delta_{ea}$ and $\delta_W$ are positive constants. This demonstrates that the Lyapunov function is UUB in both $\|\boldsymbol{e}_a\|$ and $\|\tilde{W}\|$ and thus shows Lyapunov stability of the observer in the proposed MSO method.

Once the nonlinear dynamics $\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}}$ are estimated by the neural network, an input–output feedback linearization controller given by

$$\boldsymbol{u} = \frac{1}{\boldsymbol{g_x}}\left(-\hat{\boldsymbol{\mathcal{F}}}_{\boldsymbol{x}} + \boldsymbol{f}_d - \boldsymbol{K}_p\boldsymbol{e}_1 - \boldsymbol{K}_d\boldsymbol{e}_2\right) \tag{19}$$

can be applied to the two-vector system in Eq. (8) such that the closed-loop system dynamics become

$$\dot{\boldsymbol{e}} = A_{\mathrm{CL}}\boldsymbol{e} + d \quad \text{where } A_{\mathrm{CL}} = \begin{bmatrix} 0 & 1 \\ -\boldsymbol{K}_p & -\boldsymbol{K}_d \end{bmatrix}$$
$$\text{and } d = \begin{bmatrix} 0 \\ \tilde{W}^T\phi(z) + \varepsilon \end{bmatrix} \tag{20}$$

where $d$ is bounded by $\|d\| \leq \|d_m\|$. By using the candidate Lyapunov function

$$L = \frac{1}{2}\boldsymbol{e}^T P\boldsymbol{e} \tag{21}$$

and taking its time derivative using the definition of $\dot{\boldsymbol{e}}$ in Eq. (20) and where $P$ satisfies the equation $A^T P + PA = -Q$, stability in a compact set about the origin can be shown if

$$\|\boldsymbol{e}\| \geq 2\frac{\lambda_{\max}(P)d_m}{\lambda_{\min}(Q)}$$

Though the full proof is omitted for brevity, it follows similar methods in literature, demonstrating that the Lyapunov functions in Eqs. (18) and (21) are UUB in $\|\boldsymbol{e}\|$, $\|\boldsymbol{e}_a\|$, and $\|\tilde{W}\|$ and thus shows Lyapunov stability of the proposed MSO method.

## IV. WASP Implementation

Verification and validation of the pointing system was performed through software-in-the-loop (SWIL) testing using NASA's high-fidelity Portable Object Simulation (PortOSim) flight-control simulation framework. PortOSim interfaces with the flight software and provides data as if in flight, using JPL ephemeris models in conjunction with custom models for the individual sensors and actuators aboard WASP. The simulation begins at float conditions with the centerbody uncaged and pointing at an inertial target using the heritage PID control system.

Three cases were tested in preparation for the next WASP mission, focusing on pointing performance and stability. A nominal case was run with the assumed WASP and centerbody inertia values ($J_1 = 92$, $J_2 = 572$, and $J_3 = 554$) and the disturbance environment anticipated for the mission. Each controller was tuned for the nominal case, and then tuning was held constant for two additional cases with various changes in the disturbance environment.

### A. PID Setup

The heritage PID system was tuned using historical tuning parameters such that the pointing error remains sub-arcsecond while keeping system bandwidth sufficiently low so as to not interfere with the science optics control system. The WASP bandwidth is approximated as $B_w = (\omega_n/2\pi)$, with the typical bandwidth set to $B_w = 0.4H_z$ and the damping ratio set to $\zeta = 0.9$. The final gains for all the parameters are given as approximately $K_p = 14{,}200$, $K_d = 5630$, and $K_i = 12{,}700$.

### B. SNNARC Setup

For the SNNARC algorithm, the filtered tracking error $\boldsymbol{r}$ consisted of pitch, $\theta_{e\beta}$, and yaw, $\theta_{e\gamma}$, pointing errors and the respective angular velocities $\omega_\beta$ and $\omega_\gamma$. The basis vector $\phi(z)$ was selected as $\phi(z) = [\theta_{e\beta}\ \theta_{e\gamma}\ \omega_\alpha\ \omega_\beta\ \omega_\gamma\ H_\beta\ H_\gamma]^T$, where $H_{\beta,\gamma}$ are the gimbal angles within the motor pitch/yaw hubs. The neural network weights for $\hat{W}$ and $\hat{V}$ were initialized as null matrices and consisted of 40 hidden nodes. The *sigmoid* activation function was chosen, where $\sigma(x) = (1/(1 + e^{-x}))$.

By using the gain tuning algorithms defined in Eq. (4) and selecting the design parameter $\alpha = 0$, a proportional–derivative (PD) control system is given where $K_p = (\omega_n^2/J_i^{-1})$ and $K_d = (2\zeta\omega_n/J_i^{-1})$.

The relationship between $K_v/\lambda$ and $K_p/K_d$ is given by $J_i^{-1}K_p = K_v\lambda$ and $J_i^{-1}K_d = K_v + \lambda$, where it can be shown that a real solution for $K_v$ and $\lambda$ only exists for $\zeta \geq 1$. The relationship is given by $K_v = \omega_n\zeta + \sqrt{\zeta^2 - 1}$ and $\lambda = \omega_n\zeta - \sqrt{\zeta^2 - 1}$, where $K_v = \lambda$ for $\zeta = 1$. This methodology was chosen for a more intuitive approach to gain tuning such that tuning can be understood in terms of system bandwidth. Because the damping ratio is typically near being critically damped, the tuning for when $\zeta < 1$ is given by $K_v = \lambda = \omega_n\zeta$, which gives a close approximation for the equivalent $K_p/K_d$ values. Final tuning gains for all the parameters were determined as $K_v = 5.0265$, $\lambda = 5.0265$, $M = 1.2$, $N = 1.2$, $\kappa = 0.1$, $\Theta_m = 0.01$, and $K_z = 0.001$, where the equivalent PD gains are $K_p = 14{,}000$ and $K_d = 2760$.

### C. MSO Setup

The observer states are given as $\hat{\boldsymbol{x}} = [\hat{\omega}_\alpha \quad \hat{\omega}_\beta \quad \hat{\omega}_\gamma]^T$ and when the MSO algorithm is first enabled, the estimated states are set to the current measured state vector. The MSO algorithm is set up in a similar manner to SNNARC so that a fair comparison can be made, using the same basis vector $\phi(z)$ and initializing the neural network weights $\hat{W}$ as null. For tuning of the linear controller gains, as with SNNARC, the design parameter in Eq. (4) is set as $\alpha = 0$ and gains are tuned using $\omega_n$ and $\zeta$ such that $K_p = (\omega_n^2/J_i^{-1})$ and $K_d = (2\zeta\omega_n/J_i^{-1})$. Final tuning gains for all the parameters are given as $K_p = 14000$, $K_d = 2760$, $K_2 = 10$, $\Gamma = 120$, and $\sigma = 0.001$.

## V. Simulation Results

Simulating the cases in PortOsim, the elevated disturbance environment moved the center of mass of the centerbody relative to the gimbal center such that there was a 0.25 ft-lbs imbalance along each axis, whereas for the severe disturbance environment the imbalance was 0.5 ft-lbs. Disturbances in the shaft rotation motors cause a sinusoidal disturbance in pointing, whereas disturbances in the upper atmosphere winds increase the pendulous motion imparted to the platform. Compared to the nominal case, the elevated disturbance environment doubles the magnitude of the disturbances, whereas the severe disturbance environment increases the disturbance by an order of magnitude.

### A. PID Results

For the nominal configuration, the PID control system maintained a root-mean-square (RMS) error of 0.254 arcsec in pitch ($\theta_{e\beta}$) and 0.266 arcsec in yaw ($\theta_{e\gamma}$), as seen in Fig. 1a. This performance is on a par with previous SWIL testing of other missions. In the elevated disturbance case, the RMS error increased to 0.387 arcsec in pitch and 0.402 arcsec in yaw, whereas the severe disturbance environment increased the RMS error values to 0.725 arcsec in pitch and 0.735 arcsec in yaw, as seen in Fig. 1b.

### B. SNNARC Results

In the nominal configuration, the SNNARC control system performed marginally better compared to the PID system, being able to maintain an RMS error of 0.198 arcsec in pitch and 0.212 arcsec in yaw, as seen in Fig. 2a. In the elevated disturbance case, the RMS error was 0.200 arcsec in pitch and 0.211 arcsec in yaw, whereas for the severe disturbance environment the RMS error values increased to 0.332 arcsec in pitch and 0.348 arcsec in yaw, as seen in Fig. 2b.

### C. MSO Results

In the nominal configuration, the MSO control system performed slightly better compared to the SNNARC system, being able to maintain an RMS error of 0.182 arcsec in pitch and 0.200 arcsec in yaw, as seen in Fig. 3a. In the elevated disturbance case, the RMS error actually improved to 0.162 arcsec in pitch and 0.170 arcsec in yaw, whereas for the severe disturbance environment the RMS errors remained near the nominal case at 0.201 arcsec in pitch and 0.232 arcsec in yaw, as seen in Fig. 3b.

### D. Results Discussion

Under nominal conditions, all control systems performed as expected with only marginal improvements in the performance of the SNNARC and MSO control systems as compared to PID. The expected benefit of the two methodologies, however, lies in their ability to compensate for unpredicted nonlinear dynamics. This capability is clearly seen when the disturbance environment is changed from the nominal. While the performance of the PID system began to degrade, both SNNARC and MSO maintained similar performance under elevated disturbances. The MSO algorithm actually produced a slight improvement in performance, which is likely related to increased measured dynamics leading to an improvement in the nonlinear estimation. As the disturbance environment continued to increase, the PID performance continued to degrade until the performance was no longer adequate. The SNNARC control system performance had only a marginal degradation of performance at the maximum disturbance tested, whereas the MSO algorithm had almost no degradation.

The difference in performance between the three control systems is readily apparent by examining an empirical cumulative distribution function (CDF), as seen in Fig. 4 for the pitch channel (yaw exhibits similar behavior). The empirical CDF, $F(x)$, is the proportion of the values in $x$ less than or equal to $q$, where $x$ is the pointing error in

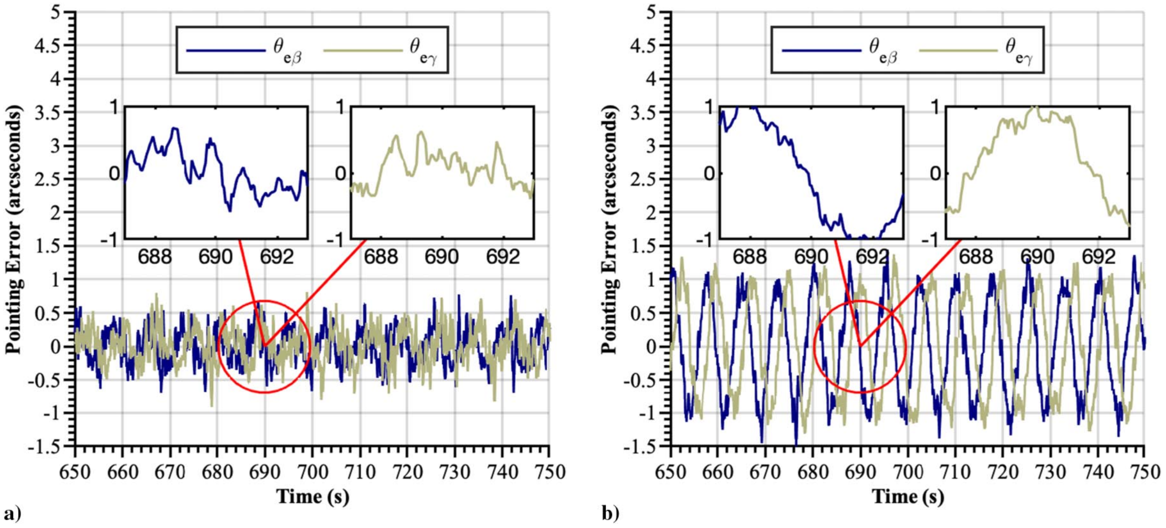

Fig. 1 Performance of the PID controller for the nominal case (a) and severe disturbance case (b).

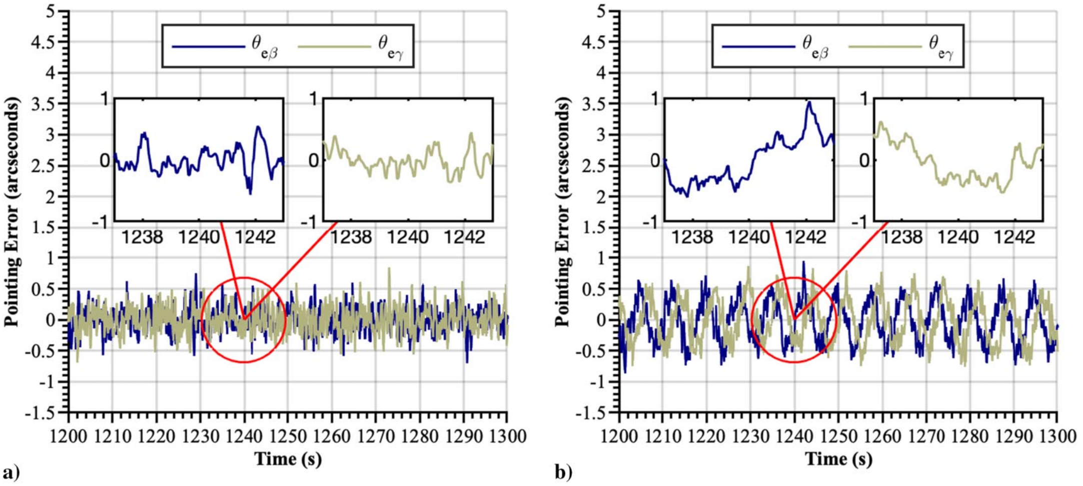

**Fig. 2 Performance of the SNNARC controller for the nominal case (a) and severe disturbance case (b).**

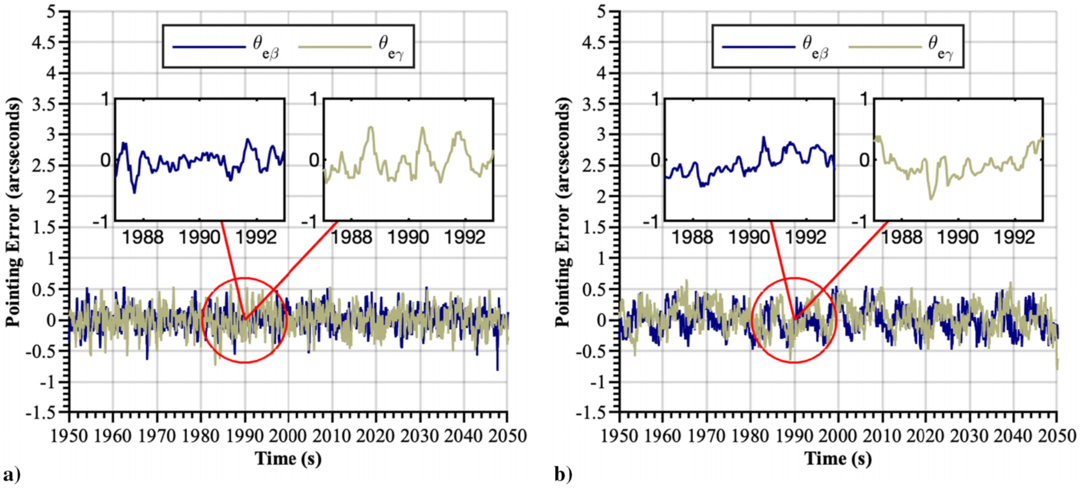

**Fig. 3 The MSO controller for the nominal case (a) demonstrated a total RMS error value of 0.270 arcsec, whereas for the severe disturbance case (b) a total RMS error value of 0.307 arcsec.**

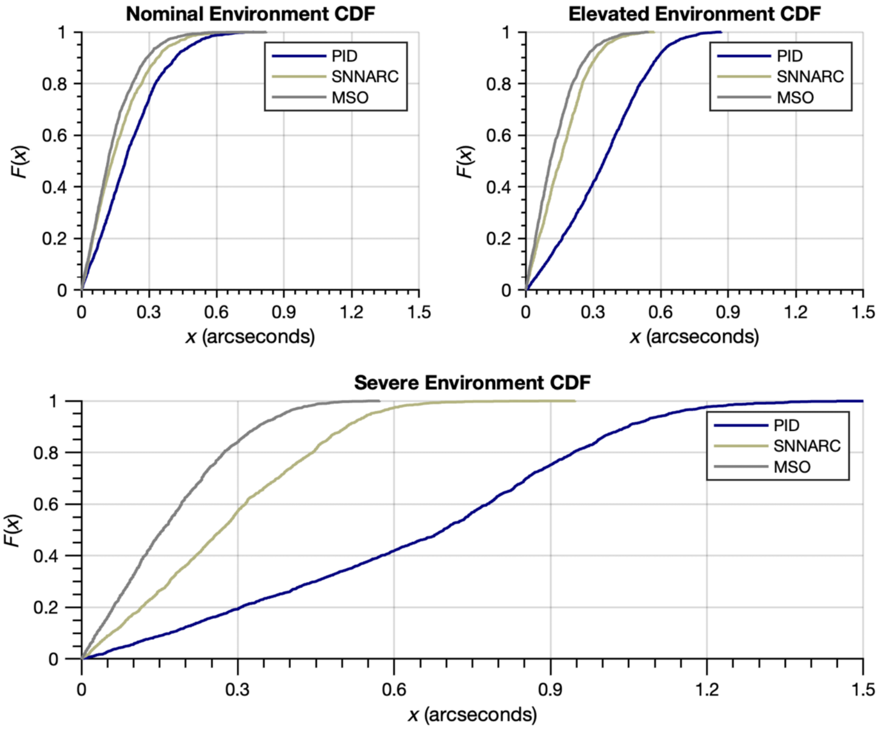

**Fig. 4 The empirical cumulative distribution function shows that MSO and SNNARC have minimal changes in pitch pointing performance, whereas the PID control system degrades quickly.**

arcseconds and $q$ is a value in $x$. It can be seen that the CDF is nearly constant for the MSO case and changes minimally for SNNARC as the disturbance environment is increased. However, the CDF moves significantly to the right for the PID system, illustrating the degradation in pointing performance.

While it is understood that the neural network learning leads to significant improvements in pointing as compared to the PID system, the differences between the two neural network methods are a little more nuanced. The MSO method is at a slight advantage as its estimation occurs outside the control space and is able to estimate at all three angular velocity channels and their dynamics, whereas the SNNARC method can only estimate the disturbances in the control channels. Along with this extra channel, the estimation through an observer produces a filtering effect as the tracking error does not drive the learning. Together, this has given the MSO method a slight advantage in pointing performance as compared to SNNARC.

## VI. Conclusions

This research has implemented two new control methodologies to the WASP system for high-altitude balloon (HAB) platforms. The SNNARC methodology used a filtered tracking error and a two-layer neural network to estimate nonlinear dynamics within the control space. The MSO methodology used an observer and a one-layer neural network to estimate the nonlinear dynamics outside the control space and then used input–output feedback linearization to control the system. Lyapunov-based proofs show that both control systems were ultimately upper bounded. Software-in-the-loop simulations were conducted using the NASA PortOSim simulation environment, testing the heritage PID control system and the two new methodologies in a nominal case and two disturbance cases. It was found that, although all three performed similarly for the nominal case, the two new methodologies were able to maintain their performance, whereas the PID system performance degraded significantly as disturbances increased. This research has shown that the addition of a neural network to a gimbal-controlled inertial pointing system suspended from a HAB platform can provide improved performance as compared to traditional heritage pointing systems.

## Acknowledgments

This research has been supported in part by the NASA Pathways Program and NASA Wallops Flight Facility. Special thanks are extended to Balakrishnan, Jim Lanzi, Scott Heatwole, and Zach Peterson for their mentorship, support, and contributions to applying this research to NASA spaceflight systems.

## References

[1] Garde, G. J., and Fairbrother, D. A., "The NASA Balloon Program—Positioning for the Future," *2015 AIAA Balloon Systems Conference*, AIAA, Reston, VA, June 2015, pp. 1–8.
https://doi.org/10.2514/6.2015-2907

[2] Fairbrother, D. A., "2017 NASA Balloon Program Update," *2017 AIAA Balloon Systems Conference*, AIAA, Reston, VA, June 2017, pp. 1–8.
https://doi.org/10.2514/6.2017-3090

[3] Stuchlik, D. W., "The Wallops Arc Second Pointer—A Balloon Borne Fine Pointing System," *2015 AIAA Balloon Systems Conference*, AIAA, Reston, VA, June 2015, pp. 1–15.
https://doi.org/10.2514/6.2015-3039

[4] Stuchlik, D. W., and Lanzi, R., "The NASA Wallops Arc-Second Pointer (WASP) System for Precision Pointing of Scientific Balloon Instruments and Telescopes," *2017 AIAA Balloon Systems Conference*, AIAA, Reston, VA, June 2017, pp. 1–8.
https://doi.org/10.2514/6.2017-3609

[5] Gopalswamy, N., Newmark, J., Yashiro, S., Mäkelä, P., Reginald, N., Thakur, N., Gong, Q., Kim, Y. H., Cho, K. S., Choi, S. H., Baek, J. H., Bong, S. C., Yang, H. S., Park, J. Y., Kim, J. H., Park, Y. D., Lee, J. O., Kim, R. S., and Lim, E. K., "The Balloon-Borne Investigation of Temperature and Speed of Electrons in the Corona (BITSE): Mission Description and Preliminary Results," *Solar Physics*, Vol. 296, Jan. 2021, pp. 1–32.
https://doi.org/10.1007/s11207-020-01751-8

[6] Mendillo, C. B., Hewawasam, K., Martel, J., Potter, T., Cook, T. A., and Chakrabarti, S., "The PICTURE-C Exoplanetary Imaging Balloon Mission: First Flight Preparation," *Techniques and Instrumentation for Detection of Exoplanets IX, Proceedings of the 2021 International Society for Optics and Photonics Conference*, Vol. 11823, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, WA, Aug. 2021, pp. 125–133.
https://doi.org/10.1117/12.2594749

[7] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, 1st ed., Springer, Berlin, 2014, Chap. 3.

[8] Slotine, J. E., and Li, W., *Applied Nonlinear Control*, Prentice Hall, Upper Saddle River, NJ, 1995, pp. 422–431.

[9] Abdo, M., Vali, A. R., Toloei, A., and Arvan, M. R., "Research on the Cross-Coupling of a Two Axes Gimbal System with Dynamic Unbalance," *International Journal of Advanced Robotic Systems*, Vol. 10, Jan. 2013, p. 357.
https://doi.org/10.5772/56963

[10] Ekstrand, B., "Equations of Motion for a Two-Axes Gimbal System," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37, No. 3, 2001, pp. 1083–1091.
https://doi.org/10.1109/7.953259

[11] Liu, S., Lu, T., Shang, T., and Xia, Q., "Dynamic Modeling and Coupling Characteristic Analysis of Two-Axis Rate Gyro Seeker," *International Journal of Aerospace Engineering*, Vol. 2018, Oct. 2018, pp. 1–14.
https://doi.org/10.1155/2018/8513684

[12] Romualdez, L. J., Damaren, C. J., Li, L., Galloway, M. N., Hartley, J. W., Netterfield, C. B., Clark, P., and Massey, R. J., "Precise Pointing and Stabilization Performance for the Balloon-Borne Imaging Testbed: 2015 Test Flight," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 231, No. 4, 2017, pp. 713–727.
https://doi.org/10.1177/0954410016641451

[13] Lewis, F. L., Jagannathan, S., and Yesildirek, A., *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor and Francis, Philadelphia, PA, 1999, Chaps. 1, 2, 6.

[14] Kahlil, H. K., *Nonlinear Control*, 1st ed., Prentice Hall, Upper Saddle River, NJ, 2015, Chaps. 12, 13.

[15] Galchenko, P., Pernicka, H. J., and Balakrishnan, S. N., "Pointing System Design for the COronal Diagnostic Experiment (CODEX) Using a Modified State Observer and a Neural Network Controller," *2020 AAS/AIAA Astrodynamics Specialist Conference*, Univelt, Inc., San Diego, CA, Aug. 2020, pp. 645–660.

[16] Sharma, M., and Calise, A. J., "Neural-Network Augmentation of Existing Linear Controllers," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 1, 2005, pp. 12–19.
https://doi.org/10.2514/1.4144

[17] Rajagopal, K., Mannava, A., Balakrishnan, S., Nguyen, N., and Krishnakumar, K., "Neuroadaptive Model Following Controller Design for Non-Affine and Non-Square Aircraft Systems," *2009 AIAA Guidance, Navigation, and Control Conference*, AIAA, Reston, VA, Aug. 2009, pp. 1–21.
https://doi.org/10.2514/6.2009-5737

[18] Mannava, A., Balakrishnan, S., Tang, L., and Landers, R., "Optimal Tracking Control of Motion Systems," *IEEE Transactions on Control Systems Technology*, Vol. 20, Nov. 2012, pp. 1548–1558.
https://doi.org/10.1109/TCST.2011.2168608

[19] Harl, N., Rajagopal, K., and Balakrishnan, S. N., "Neural Network Based Modified State Observer for Orbit Uncertainty Estimation," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1194–1209.
https://doi.org/10.2514/1.55711

[20] Chandramohan, R., and Calise, A. J., "Output Feedback Adaptive Control of Plants with Unmodeled Dynamics and Input Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 10, 2019, pp. 2143–2156.
https://doi.org/10.2514/1.G003548