Design of an Active Vibration Control Platform for the MQ-9A Reaper's Raytheon MTS-B Turret

Angelos Kamariadis
School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
angeloskamariadis@gmail.com

***Abstract*—Unmanned Combat Aerial Vehicles (UCAVs), specifically the American MQ-9A Reaper, manufactured by General Atomics Aeronautical Systems, rely on highly sensitive Electro-Optical/Infrared (EO/IR) payloads for precision targeting and reconnaissance. However, these optical systems are continuously subjected to severe mechanical disturbances during missions, predominantly characterized by the Blade Pass Frequency harmonics of the turboprop propulsion system, as well as low frequency stochastic aerodynamic buffeting. This project presents the mathematical modeling and theoretical design of an Active Vibration Control platform, engineered to stabilize the Raytheon Multi-Spectral Targeting System (MTS-B), in order to prevent high zoom video feeds and laser designators from becoming useless during flight. The proposed closed loop pipeline, strictly leverages advanced Digital Signal Processing and Control Theory to mathematically neutralize aerospace mechanical noise. In order to isolate parasitic structural jitter from intentional, non linear flight kinematics, the DSP architecture employs an Extended Kalman Filter fused with dynamic notch filtering. The resulting pristine error state, is then processed by a discrete time Filtered-x Least Mean Squares (FxLMS) feedforward control algorithm, utilizing state-space representation to calculate the exact inverse counter forces required. Ultimately, this project provides a comprehensive MATLAB/Simulink simulation, for internal optical actuation, demonstrating the theoretical attenuation of high frequency disturbances to achieve near zero shaking of the camera lens.**

***Index Terms*—Active Vibration Control, Extended Kalman Filter, Filtered-X LMS Algorithm, Fast Steering Mirror, State-Space Design, Digital Signal Processing, MATLAB/Simulink, MQ-9A Reaper, MTS-B Turret, Stabilization,.**

I. Introduction

At its core, Active Vibration Control (AVC) functions as the mechanical counterpart to acoustic noise cancellation. Rather than using microphones and speakers to neutralize airborne sound waves, AVC utilizes motion sensors and mechanical actuators to actively neutralize structural vibrations in solid objects. By applying an equal and opposite physical force in real-time, the system effectively stabilizes the target platform. The AVC process operates as a rapid and continuous feedback loop. It begins when a system encounters a disturbance, such as a turbine engine or aerodynamic buffeting vibrating the UCAV's targeting payload. Immediately, sensors including high-frequency accelerometers and gyroscopes, specifically Inertial Measurement Units (IMUs), detect the shaking while capturing precise data on its frequency, direction and intensity. A microcontroller processes this telemetry to compute the exact counter movement required. A mechanical actuator, such as a voice coil or motorized piston, then executes this command by applying an equal and opposite force. Ultimately, these two physical forces, the original disturbance and the system's active counter force, collide and cancel each other out. As a result, the isolated payload appears to float completely still to an outside observer, even as the foundational base shakes violently. In short, AVC is the engineering practice of making an object hold perfectly still in a chaotic, moving environment by actively punching back against the chaos.

II. Theoretical Background

A. General Atomics MQ-9A Reaper UCAV

Fig. 1. The MQ-9A Reaper on the tarmac. This image highlights the physical separation between the disturbance source and the critical payload.

Below, all the necessary information about the MQ-9A Reaper is presented.

The Raytheon Multi-Spectral Targeting System (MTS-B) is the primary electro-optical and infrared (EO/IR) sensor suite integrated onto the baseline MQ-9A Reaper. As mentioned above, the ultimate goal of this system is to protect the Reaper's primary plant, the Raytheon MTS-B EO/IR turret, aerospace mechanical noise. If this turret shakes, the laser designator misses its target and high zoom video feeds become completely useless.

1) The Disturbance Parameters (Engine and Propeller): In order to simulate the high-frequency noise mathematically, we will need the exact specs of the propulsion system, which are:

- **Engine:** Honeywell TPE331-10 turboprop (940 Shaft Horsepower).
- **Power:** 45 kVA. 45 kVA is a massive amount of onboard electrical power so the MQ-9A has more than enough excess electrical generation to power high-speed Digital Signal Processors (DSPs), Analog-to-Digital Converters (ADCs), and the linear amplifiers required to drive the internal Fast Steering Mirrors (FSMs) without straining the aircraft's critical flight systems.
- **Backup Power:** 4kVA

We will need to calculate the exact BPF. The Reaper's engine typically drives a 3-blade constant-speed propeller. While turbine RPM is extremely high (often ¿ 40000 RPM), the gearbox reduces this to a constant propeller speed (often around 1591 RPM in flight).

$$f_{prop} = \frac{1591}{60} \approx 26.5 \text{ Hz}$$
$$BPF = f_{prop} \times 3 \text{ (blades)} \approx 79.5 \text{ Hz}$$

Giving the above, for this project we will assume a BPF of 80 Hz.

2) The Plant Parameters (Payload): Our State-Space matrix (A, B, C, D) needs the mass and dimensions of what we are trying to stabilize.

- **Payload Capacity:** The Reaper has a 1701 kg external and 386 kg internal payload capacity, but you only care about the centerline chin hardpoint.
- **The Target Sensor:** The Raytheon Multi-Spectral Targeting System (MTS-B / AN/DAS-1).
- **Mass Constraint:** The MTS-B turret weighs approximately 54 kg.
- **Application:** This confirms the logic behind Stage 5, of the control pipeline presented later. We cannot mathematically vibrate a 54 kg sphere at 80 Hz using small internal motors. This proves why our mathematical model must focus on actuating the tiny, lightweight Fast Steering Mirrors (FSMs) inside the turret and not the turret housing itself.
- **7 external hardpoints (6 wing-mount and 1 centerline):** The Raytheon MTS-B turret, is mounted on that exact centerline station under the chin. This suggests that the payload is situated perfectly on the aircraft's roll axis but is highly susceptible to pitch and yaw vibrations traveling down the center of the fuselage.

3) The Flight Dynamics and The Dimensions of the Vehicle (Low-Frequency Noise): To simulate the stochastic aerodynamic buffeting and structural flexure, we will need the aircraft's performance envelope, which is presented below:

- **Max True Airspeed:** 240 KTAS (Knots True AirSpeed).
- **Operating Altitude:** Up to 13716 meters.
- **Wingspan:** Approximately 20 meters. A 20 meter wingspan creates massive, low-frequency structural bending modes, often between 1 Hz and 5 Hz. Additionally, flying at 240 KTAS creates a chaotic broadband aerodynamic noise profile. In our disturbance model, we will inject low-frequency Gaussian white noise to represent this buffeting, forcing the Extended Kalman Filter to separate the 2 Hz wing flex from the 80 Hz engine vibration.
- **Length:** 11 meters. Because the engine is at the extreme rear and the camera is at the extreme front, the vibration must travel through 36 feet of aluminum and composite fuselage. In our feedforward control loop (FxLMS), this physical distance creates a measurable propagation delay (phase shift) that our DSP must account for when calculating the counter-force.

While fuel consumption continuously alters the MQ-9A Reaper's overall mass and center of gravity over a 24-hour mission, modeling this fuel burn is intentionally excluded from the project scope due to a massive separation in time scales. Fuel depletion is a slow, macroscopic dynamic occurring over hours, whereas the active vibration control (AVC) loop operates at microscopic timescales, executing signal processing and actuation corrections at a rate of 2 kHz to 4 kHz. On a millisecond by millisecond basis, the system's mass matrix remains effectively static, allowing the simulation to prioritize high-frequency adaptive filtering and loop stability without introducing unnecessary and slow-moving kinematic variables.

B. Digital Signal Processing

The Digital Signal Processor serves as the mathematical brain of the system.

1) Nyquist - Shannon Sampling Theorem:

$$f_s \geq 2f_{max}$$

where f_s is our sampling frequency (how many times per second the Analog-to-Digital Converter reads the sensor) and f_{max} is the highest frequency component existing in the analog signal.

Before any math can happen, the continuous analog signal from the Reaper's IMUs must be digitized. The foundational law of Digital Signal Processing is the Nyquist-Shannon sampling theorem. It states that to perfectly reconstruct a signal, your sampling rate f_s must be strictly greater than twice the highest frequency component f_{max} present in the signal. If high-frequency engine harmonics exceed the Nyquist limit $f_s/2$, they will fold back into the lower frequencies, creating phantom low-frequency vibrations, a catastrophic error called aliasing.

2) The Z Transform: In classical continuous-time control, we use the Laplace transform. But because our system is completely digital, we must work in the discrete-time domain using the equivalent to the Laplace transform. The Z-transform which converts difference equations into algebraic equations, allowing us to analyze system stability. The transform of a discrete-time signal x[n] is defined as:

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, z = re^{j\omega}$$

3) Digital Filtering: In order to strip away broadband aerodynamic noise and isolate the turboprop's Blade Pass Frequency, we will mathematically filter the data using one of two primary architectures:

- **Finite Impulse Response (FIR):** Which is always stable, it has linear phase which means it does not distort the signal in time, and is computationally heavy. In Active Vibration Control, FIR filters are used in precision phase-matching for feedforward control loops.

FIR filters calculate their output entirely on the basis of the current and past values of the input signal. They do not contain feedback loops. In the discrete time domain, the output y[n] is simply the convolution of the input signal x[n] with a finite array of filter coefficients b_k.

$$y[n] = \sum_{k=0}^{N-1} b_k x[n-k]$$

Where N is the total number of coefficients and the length of the filter. In order to achieve a sharp cutoff, N must be very large, making FIR filters computationally heavy. By converting the difference equation into the complex domain, we obtain the transfer function of the filter.

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k}$$

To determine system stability, we must evaluate the poles of the transfer function. By factoring out the highest power of z, the transfer function can be rewritten as a rational function:

$$H(z) = \sum_{k=0}^{N-1} b_k z^{-k} = \frac{b_0 z^{N-1} + b_1 z^{N-2} + \cdots + b_{N-1}}{z^{N-1}}$$

Because the denominator is strictly z^{N-1}, all $N-1$ poles of the FIR filter are located exactly at $z = 0$ in the complex plane. Since the stability criterion dictates that all poles must strictly lie inside the unit circle $|z| < 1$, an FIR filter is mathematically incapable of diverging or becoming unstable.

Furthermore, FIR filters can achieve strict linear phase by forcing their coefficients to be perfectly symmetrical or anti-symmetrical around the center point:

$$b_k = \pm b_{N-1-k}$$

This symmetry, ensures that the phase response is a strictly linear function of frequency, meaning all frequencies of the turboprop vibration are delayed by the exact same constant group delay. This prevents wave distortion from ruining the FxLMS phase-matching process.

- **Infinite Impulse Response (IIR):** Which is highly efficient, but introduces phase lag and the potential to become unstable. In Active Vibration Control, IIR Filters are used in aggressive low-pass filtering to block high-frequency sensor noise.

IIR filters are recursive. They calculate their output using both past inputs and past outputs. This feedback mechanism allows them to act like physical analog circuits. The output y[n] requires two sets of coefficients, b_k for the feedforward input paths, and a_k for the recursive feedback paths.

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] - \sum_{k=1}^{N} a_k y[n-k]$$

IIR filters are highly efficient. The recursive $y[n-k]$ terms allow a 4^{th}-order IIR filter to achieve the same steep frequency cutoff that would require a 100th-order FIR filter, drastically reducing the Digital Signal Processor's workload.

Once again, by converting the difference equation above into the complex domain, we obtain the transfer function of the filter.

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

Mathematical Risk for Instability: Unlike FIR filters, the denominator of an IIR filter is a complex polynomial. The roots of this specific polynomial dictate the location of the filter's poles. If aggressive filtering is required, or if the hardware A/D quantization noise shifts the coefficients slightly, a pole can be mathematically forced outside the unit circle. If this occurs, the recursive feedback loop will mathematically amplify to infinity, causing the physical active vibration control system to catastrophically diverge.

4) Analog-to-Digital Converter: In any modern electromechanical control system, Analog-to-Digital Converters (ADCs) serve as the vital bridge between the continuous physical environment and the discrete mathematical domain. For the MQ-9A Reaper's AVC pipeline, ADCs are a fundamental necessity for enabling digital computation. Their role is:

- **Bridging the Continuous and Discrete Domains:** Physical reality, such as the 80 Hz mechanical waves generated by the turboprop and the subsequent voltage output from the piezoelectric or MEMS sensors, is continuous in both time and amplitude. However, the system's Digital Signal Processor, operates exclusively in the discrete time domain, computing in binary. The DSP is mathematically blind to continuous analog voltage.
- **Enabling Advanced Algorithms:** Extended Kalman Filter (EKF) and Filtered-x Least Mean Squares (FxLMS) consist of algebraic difference equations requiring discrete data arrays. ADCs are responsible for taking rapid snapshots of the continuous sensor voltage at precise intervals and quantizing those voltages into digital numerical values that the DSP can mathematically manipulate.
- **Preserving Micro-Dynamics:** Without an ADC of sufficient resolution, the subtle high frequency structural jitters would be lost. By slicing the continuous analog signal into thousands of highly precise digital steps, a high resolution ADC ensures that the micro vibrations

degrading the optical path are faithfully represented in the data without being drowned out by mathematical quantization noise.
Ultimately, without an ADC the continuous mechanical disturbance can never be translated into the discrete numbers required for the control system to calculate the necessary counter force.

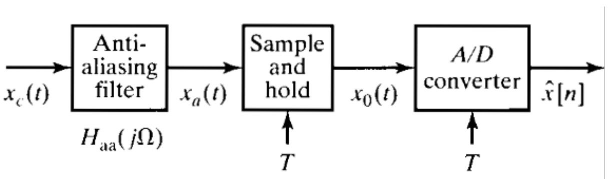


Fig. 2. Block Diagram of Analog Front-End (Last step is the ADC).

- **Anti-Aliasing Filter:** A low-pass filter which cleans the continuous physical disturbance before it is sampled, ensuring that high frequency mechanical noise does not alias and violate the Nyquist-Shannon theorem. It allows all the low-frequency signals we actually care about to pass through to the ADC, while blocking any high frequency noise that violates the Nyquist-Shannon sampling theorem.
The following equation defines the effective analog low-pass filter, based on our sampling period. This mathematically represents the ideal physical anti-aliasing filter, passing analog frequencies below ω_c/T, while blocking everything else.

$$H_{aa}(j\Omega) = \begin{cases} 1, \forall|\Omega| < \frac{\omega_c}{T} \\ 0, \forall|\Omega| > \frac{\omega_c}{T} \end{cases}$$

$$x_a(t) = x_c(t) * h_{aa}(t) = \int_{-\infty}^{\infty} x_c(\tau) h_{aa}(t-\tau) d\tau$$

$$\equiv X_a(j\Omega) = X_c(j\Omega) \cdot H_{aa}(j\Omega)$$
$$\Rightarrow x_a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_c(j\Omega) \cdot H_{aa}(j\Omega) e^{j\Omega t} d\Omega$$

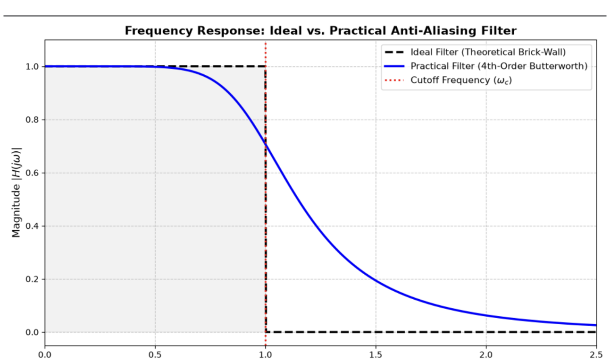


Fig. 3. Ideal vs Practical Anti-Aliasing Filter Frequency Response.

- **Sample and Hold:** Freezes a snapshot of that analog voltage at precise discrete intervals

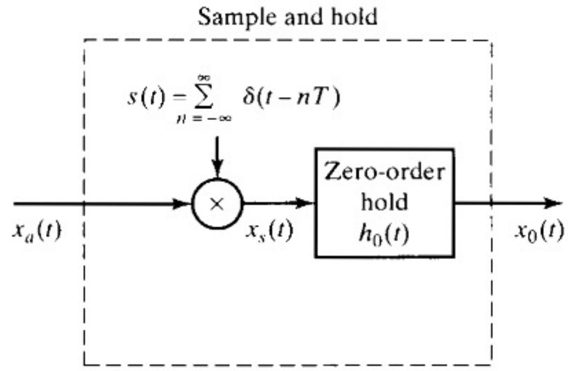


Fig. 4. Representation of ideal sample-and-hold.

The ideal sample-and-hold system has output:

$$x_0(t) = \sum_{n=-\infty}^{\infty} x[n] h_0(t - nT)$$

$$\text{where } x[n] = x_a(nT) \text{ and } h_0(t) = \begin{cases} 1, t \in (0, T) \\ 0, otherwise \end{cases}$$

$$\Rightarrow x_0(t) = h_0(t) * \sum_{n=-\infty}^{\infty} x_a(nT) \delta(t - nT)$$

- **A/D Converter:** A device that converts analog voltage and mathematically quantizes it into the discrete digital arrays (binary code) which is required by the DSP, in order to run the FxLMS algorithm. Before the converter can output a value, it must define the physical voltage size of a single binary step (the LSB), using the step function following.

Before the converter can output a value, it must define the physical voltage size of a single binary step (the LSB), using the step function following.

$$\Delta = \frac{2X_m}{2^N}$$

The input to the A/D converter is the frozen voltage from the sample-and-hold circuit, $x_0(t)$. At the exact instant of conversion ($t = nT$), this held voltage is mathematically identical to the continuous anti-aliased signal $x_a(t)$.

$$x_0(nT) = x_a(nT)$$

This is the exact mathematical operation the A/D silicon executes. It takes the continuous frozen voltage, divides it by the step size, rounds it to the nearest integer using a floor function, and scales it back. This is known as a mid-tread uniform quantization function:

$$\hat{x}[n] = \Delta \cdot \left\lfloor \frac{x_a(nT)}{\Delta} + 0.5 \right\rfloor$$

We know that $x[n] = x_a(nT)$ and thus, we obtain:

$$\hat{x}[n] = \Delta \cdot \left\lfloor \frac{x[n]}{\Delta} + 0.5 \right\rfloor$$

In general, the quantized sample $\hat{x}[n]$ will differ from the true sample value $x[n]$. The difference between them is the quantization error, defined as:

$$e[n] = \hat{x}[n] - x[n] \in \left(-\frac{\Delta}{2}, \frac{\Delta}{2} \right]$$

To the Digital Signal Processor, the output is modeled strictly as the true physical signal corrupted by additive, uniformly distributed quantization noise. Thus:

$$\hat{x}[n] = x_a(nT) + e[n]$$

In terms of the quantization error $e[n]$, we assume that succesive noise sapmles are uncorrelated with each other and that $e[n]$ is also uncorrelated with $x[n]$. Thus, $e[n]$ is assumed to be a uniformly distributed white noise sequence. The mean value of $e[n]$ is zero and its variance is:

$$\sigma_e^2 = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12}$$

For a (B + 1)-bit quantizer with full scale value $X_m \Rightarrow \sigma_e^2 = \frac{X_m^2 \cdot 2^{-2N}}{3}$

A common measure of the amount of degradation of a signal by additive noise is the signal to noise ratio, defined as:

$$SNR = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) = 10 \log_{10} \left(\frac{\sigma_x^2}{\frac{2^{-2N} X_m^2}{12}} \right)$$

$$\Rightarrow SNR = 6.02B + 10.8 - 20 \log_{10} \left(\frac{X_m}{\sigma_x} \right)$$

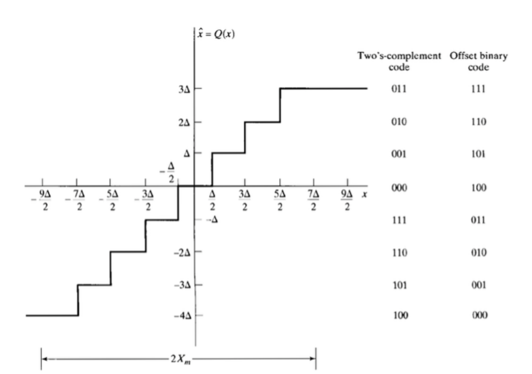


Fig. 5. Typical quantizer for A/D conversion

The provided figure above, illustrates the input-output transfer characteristic of a uniform, mid-tread quantizer, visually representing the exact mathematical rounding executed by the A/D Converter. The horizontal axis represents the frozen, continuous input voltage x from the sample-and-hold circuit, while the vertical axis displays the resulting discrete digital output $\hat{x}[n] = Q(x)$. The characteristic staircase geometry demonstrates the quantization process. Any continuous voltage falling within a specific interval of width Δ is mathematically snapped to a single horizontal discrete level. Because this is a mid-tread architecture, there is a flat step perfectly centered at the origin, ensuring that a true zero-voltage input (a perfectly still UCAV turret) translates strictly to a digital zero, which prevents the control algorithm from attempting to correct artificial micro-fluctuations. The binary code in right illustrates how the hardware translates physical levels into the binary arrays fed directly into the Digital Signal Processor.

C. Control Theory

Once DSP delivers a mathematically pristine representation of the vibration, Control Theory dictates how to calculate the counter-force.

1) State-Space Representation (Modern Control: For a multi-variable aerospace system, standard transfer functions are insufficient. We will model the dynamics of the Raytheon MTS-B turret using discrete-time State-Space representation. This uses matrices to define the internal state, inputs, and outputs of the system:

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

Where $x(t)$ is the state vector (position, velocity), $u(t)$ is the input vector (the counter-vibration voltage commands), $y(t)$ is the output vector (what our sensors measure), A is the system matrix which defines the physical and structural resonance of the turret, B is the input matrix which defines how the actuator voltage forces the system to change, and lastly, C the output matrix maps the internal physical states to the specific IMU sensor outputs. The Feedthrough Matrix usually is assumed to be zero in mechanical systems, representing direct input-to-output bypassing.

2) Feedback Control (Reactive): Feedback control measures the error at the camera lens and tries to push it to zero. Algorithms like PID (Proportional-Integral-Derivative) or LQG (Linear Quadratic Gaussian) are highly effective at suppressing the low-frequency, unpredictable aerodynamic buffeting. For instance, the standard PID controller generates its corrective command, $u(t)$, based on the measured error signal, $e(t)$, using proportional, integral, and derivative gains (K_p, K_i, K_d):

$$u(t) = K_p e(t) + K_i \int_0^t e(t) d\tau + K_d \frac{de(t)}{dt}$$

However, feedback is inherently reactive; it only acts after the camera has already started shaking. The processing and physical actuation delays make it far too slow to cancel the deterministic, high-frequency 80 Hz turboprop vibration. To neutralize this structural resonance before it degrades the image, the architecture must incorporate a proactive, feedforward control path.

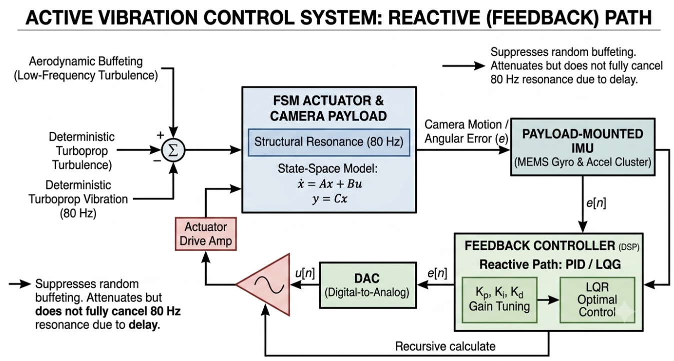


Fig. 6. Block diagram of the reactive (feedback) control architecture, illustrating the PID/LQG loop used to suppress low-frequency aerodynamic buffeting.

3) Feedforward Control (Proactive): To effectively cancel deterministic, high-frequency engine harmonics, the system necessitates a feedforward control architecture. By positioning a high-bandwidth reference sensor directly on the UCAV fuselage, the digital signal processor detects the primary structural vibration, $x(n)$, before it physically propagates to the camera payload. The algorithm calculates the physical propagation delay and structural dynamics modeled as the primary path, $P(z)$ and commands the Fast Steering Mirrors (FSMs) to actuate simultaneously with the incoming wave.

Mathematically, if the secondary path (the physical transfer function from the actuator to the error sensor) is defined as $S(z)$, the ideal digital feedforward controller, $W(z)$, aims to achieve perfect destructive interference at the lens by satisfying the following optimal transfer function:

$$W(z) = -\frac{P(z)}{S(z)}$$

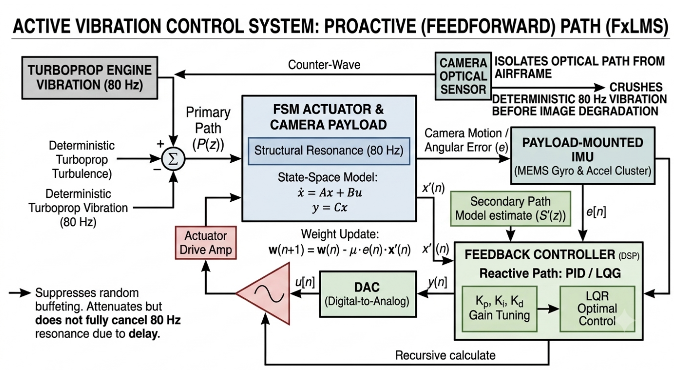


Fig. 7. Block diagram of the proactive (feedforward) control architecture utilizing the FxLMS algorithm to preemptively neutralize the 80 Hz structural resonance.

By passing the reference signal through this filter, the controller generates a precise, phase-inverted command that physically neutralizes the disturbance the exact millisecond it arrives. However, because the UCAV's structural dynamics and acoustic environment change dynamically during flight due to changing airspeeds and fuel loads, a static filter $W(z)$ is insufficient. This physical reality directly necessitates an adaptive architecture, such as the FxLMS algorithm, to continuously calculate and update these filter weights in real-time.

4) The FxLMS Algorithm: The absolute core of this project, is the Filtered-x Least Mean Squares (FxLMS) algorithm. It is an adaptive feedforward controller. Standard LMS fails in physical systems because the actuator and amplifier introduce their own delay and phase shift. FxLMS solves this by passing the reference signal through a mathematical model of the secondary path, denoted as $S(z)$, before updating the filter weights.

To establish a complete mathematical foundation, the physical signal path must first be defined. The adaptive filter generates the counter-vibration command $y(n)$ by computing the inner product of the current filter weights $\mathbf{w}(n)$ and the reference signal vector $\mathbf{x}(n)$:

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$$

The physical error measured at the camera lens, $e(n)$, is the superposition of the primary turboprop disturbance $d(n)$ and the generated anti-noise $y(n)$ after it propagates through the true physical secondary path $s(n)$:

$$e(n) = d(n) - s(n) * y(n)$$

To accurately compensate for this physical propagation delay, the reference signal is convolved with the mathematical estimate of the secondary path impulse response, $\hat{s}(n)$, creating the filtered reference signal $x'(n)$:

$$x'(n) = \hat{s}(n) * x(n)$$

By utilizing this filtered reference vector $\mathbf{x}'(n)$, the FxLMS algorithm dynamically updates its filter weights according to the following equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}'(n)$$

where μ is the step-size. The mean-square-error (MSE) of the FxLMS algorithm, $J(n)$, is expressed as:

$$J(n) = J_o + \sum_{q=0}^{Q-1} s_q^2 E\left\{\mathbf{c}^T(n-q)\mathbf{\Lambda}\mathbf{c}(n-q)\right\}$$

where J_o is the minimal MSE level, $\mathbf{\Lambda}$ is the diagonalized auto-correlation matrix of $x(n)$, s_q denotes elements of the secondary path vector $\mathbf{s}$, Q is the length of $\mathbf{s}$, and $\mathbf{c}(n)$ is the deviation of $\mathbf{w}(n)$ from its optimal vector. The excess-MSE function, $J_{ex}(n)$, is defined as a dynamic measure determining the deviation of $J(n)$ from its minimal level, and thus we obtain:

$$J_{ex}(n) = \sum_{q=0}^{Q-1} s_q^2 E\left\{\mathbf{c}^T(n-q)\mathbf{\Lambda}\mathbf{c}(n-q)\right\}$$

5) The Extended Kalman Filter: While classical digital filtering operates strictly within the frequency domain to attenuate specific bandwidths, the Kalman filter functions in the time domain as an optimal state estimator. It utilizes a mathematical model of the system to predict theoretical sensor readings, compares them against actual measurements, and dynamically computes the most statistically probable state. Because the 3D rotational kinematics of the MQ-9A Reaper are inherently non-linear, standard Kalman filtering is insufficient. Therefore, an Extended Kalman Filter (EKF) is deployed to compute Jacobian matrices, mathematically linearizing the system at every discrete time step.

To execute the recursive predict-update loop effectively, the physical system is governed by two primary non-linear functions. The state transition function, $f(\hat{\mathbf{x}}_k)$, predicts the UCAV's 3D rotational kinematics for the subsequent time step. Concurrently, the measurement model, $h(\hat{\mathbf{x}}_k^-)$, maps these internal kinematic states to the precise outputs expected from the payload-mounted IMUs. Because standard matrix operations cannot be directly applied to non-linear equations, the EKF calculates the Jacobians $\mathbf{F}_k$ (the State Transition Jacobian) and $\mathbf{H}_k$ (the Measurement Jacobian). These matrices compute the partial derivatives of the non-linear curves at that exact millisecond, temporarily forcing the system into a linear state-space representation.

Furthermore, this optimal estimation relies on rigorously defining the statistical uncertainty of the system via two covariance matrices:

- **$\mathbf{R}$ (Measurement Noise Covariance):** This matrix models the baseline uncertainty of the physical sensors. Critically, it encapsulates the uniform quantization noise, $e[n]$, introduced by the Analog-to-Digital Converter, as well as the inherent electronic noise floor of the IMUs.
- **$\mathbf{Q}$ (Process Noise Covariance):** This matrix defines the uncertainty within the mathematical physics model itself, accounting for unmodeled mechanical disturbances, such as unpredictable aerodynamic buffeting pushing the UCAV off its projected trajectory.

With the system statistics defined, the EKF executes its recursive loop. During the Measurement Update (Correction) phase, the filter calculates the Kalman Gain, $\mathbf{K}_k$, which dictates how heavily the algorithm should trust the new sensor measurement, $\mathbf{z}_k$, versus its own mathematical prediction:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1}$$

The optimal state estimate, $\hat{\mathbf{x}}_k$, is then updated using the calculated gain and the measurement residual:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-))$$

Finally, the algorithm updates $\mathbf{P}_k$, the Error Covariance Matrix. $\mathbf{P}_k$ represents the filter's statistical certainty in its own state estimates. As the recursive loop ingests valid sensor data, $\mathbf{P}_k$ mathematically shrinks:

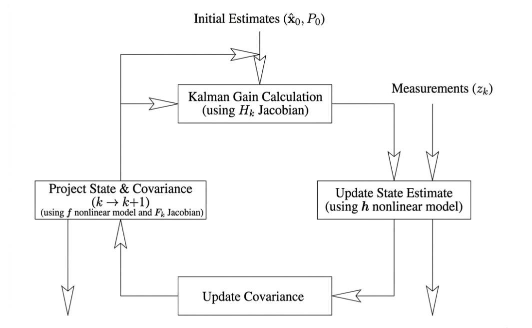
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$$

Once the current state is corrected, the algorithm executes the State and Covariance Prediction phase for the next time step ($k \to k+1$), projecting the kinematics forward through the non-linear physics model and Jacobian transition matrix:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}$$

The overall architecture of this recursive algorithm is illustrated in figure below. The EKF operates in a continuous, two-step feedback loop. The upper block represents the Measurement Update, where real-time IMU data is ingested to correct the previous predictions. The lower block represents the State and Covariance Prediction, where the non-linear physics model projects the corrected state forward to the next time step. The exact mathematical execution of this loop is governed by the equations detailed below.



Description	Equation
Kalman Gain	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1}$
Update State Estimate (uses nonlinear function h)	$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-))$
Update Covariance	$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^-$
State & Covariance Prediction ($k \to k+1$) (uses nonlinear function f and Jacobian F_k)	$\hat{\mathbf{x}}_{k+1}^- = \mathbf{f}(\hat{\mathbf{x}}_k)$ $\mathbf{P}_{k+1}^- = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^T + \mathbf{Q}$

Jacobian Definitions: $\mathbf{F}_k = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k}$ (State Transition Jacobian) $\mathbf{H}_k = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k^-}$ (Measurement Jacobian)

Fig. 8. The recursive predict-update architecture and matrix equations of the Extended Kalman Filter.

6) Systems Stability: In control theory, proving our math works means proving it will not explode into infinite oscillation.

- **Pole Placement:** In the Z-domain, a system is only stable if all its poles which are the roots of the denominator of its transfer function, lie inside the unit circle on the s-plane ($|z| < 1$).
- **The Cost of Delay:** In active vibration control, computational latency is our greatest enemy. If our Digital Signal Processing math take too long to compute, the actuator will push with the vibration wave instead of against it, causing catastrophic constructive interference.

III. Control Pipeline

Unlike passive isolation which only works well at high frequencies and can actually amplify low-frequency resonance, an active system uses sensors, compute and actuators to dynamically cancel out vibrations in real-time.

Building a functional Active Vibration Control (AVC) system requires designing a real-time, closed-loop electromechanical pipeline. Rather than viewing this merely as physical movement, it is best understood as a continuous transformation of energy and data: Physical → Analog → Digital → Analog → Physical.

UCAVs like the American MQ-9A Reaper rely on highly sensitive Electro-Optical/Infrared (EO/IR) camera gimbals and laser designators. If those sensors vibrate, a target 5 kilometers away becomes a blurry smudge, and a laser designator misses its mark.

Since we obviously cannot get our hands on a classified military asset, we will need to build a sub-scale technology demonstrator.

Below, lies the 6-stage DSP and control pipeline, of the AVC platform for the MQ-9A Reaper UCAV.

A. Stage 1: The Disturbance

The process begins with mechanical energy entering our system. On a defense platform like the American MQ-9A Reaper, this mechanical disturbance is heavily dominated by the aircraft's Honeywell TPE331-10 turboprop engine. Operating at high shaft horsepower with a constant propeller RPM, the propulsion system generates a massive, continuous harmonic vibration specifically known as the Blade Pass Frequency (BPF). This high-frequency mechanical energy travels directly down the fuselage and reaches the Raytheon Multi-Spectral Targeting System (MTS-B) EO/IR turret, which serves as our plant. At this moment, the sensitive optical payload experiences an unwanted, highly dynamic acceleration that severely degrades targeting accuracy. The vibration profile on a fixed-wing UCAV like the MQ-9A is entirely different from a standard quadcopter, and includes the following specific disturbances: **Engine Harmonics (Blade Pass Frequency):** A high-frequency, continuous harmonic vibration generated by the constant rotation of the turboprop propeller, creating predictable but intense mechanical noise. **Aerodynamic Buffeting:** Chaotic, low-frequency stochastic noise caused by the airframe slicing through the atmosphere at cruising speeds up to 240 KTAS. **Structural Flexure:** Low-frequency resonant bending modes generated by the Reaper's long, high-aspect-ratio wings, which continuously shift the baseline axis of the fuselage in flight.

B. Stage 2: Sensing and Digitization

Prior to neutralizing the movement, the system must measure the disturbance with extremely low latency. To achieve this on the MQ-9A Reaper, the architecture requires a dual-sensor setup. A primary reference sensor, a tactical-grade MEMS IMU, is rigidly mounted to the vehicle's fuselage to detect the turboprop's vibration before it reaches the payload. Simultaneously, an error-sensing IMU is mounted directly onto the Raytheon MTS-B turret to measure the residual acceleration.

These sensors generate continuous analog voltages proportional to the structural shifting, which are subsequently quantized by high-resolution Analog-to-Digital Converters (ADCs). To prevent aliasing when dealing with the high-frequency engine harmonics, the ADCs' sampling rates must strictly adhere to the Nyquist-Shannon sampling theorem.

In this specific acoustic environment, the ADCs must operate at a minimum of 2 kHz to 4 kHz, well over twice the highest frequency of the turboprop's vibration, ensuring the digital signal processor receives a flawless, real-time representation of the mechanical chaos.

The Consequence of Failing: Aliasing: If our ADC samples too slowly ($f_s < 2f_{\text{max}}$), we don't just miss the high-frequency data, but actually we corrupt the low-frequency data. This phenomenon is called aliasing. When a vibration frequency exceeds the Nyquist frequency, the digital system misinterprets it. The high-frequency wave folds back down into the digital data and masquerades as a low-frequency wave. A useful analogy is the Helicopter Blade Illusion, where the rotor blades of a helicopter appear to be moving in slow motion, while the observer is standing completely still, or rotating backward. This is visual aliasing. The camera's frame rate (sampling rate) is too slow to capture the true speed of the blades, creating a false digital reconstruction of reality. If this happens in an AVC system, our microcontroller will think the platform is vibrating slowly to the left, when it is actually vibrating rapidly to the right. So, our actuator will push in the wrong direction, amplifying the vibration instead of canceling it.

In practice, simply meeting the bare minimum of $2f_{\text{max}}$ is a bad idea, especially in control systems, because it introduces phase lag. In order to handle this in our design, we must first determine our target. If the maximum vibration we expect from our mechanical disturbance (like a rumble motor) is for instance 100 Hz, our absolute theoretical minimum sampling rate is 200 Hz. In practice, engineers usually oversample by a factor of 10 to 20 to ensure minimal latency and smooth control loops. So, we would want to sample that 100 Hz vibration at 1000 Hz or 2000 Hz. Finally, in order to guarantee that no rogue high-frequency noise like electrical static or high-pitch resonance sneaks into your ADC and causes aliasing, we must place an analog Low-Pass Filter (LPF) before the signal reaches the ADC. This physically blocks any frequencies higher than your Nyquist limit from ever entering the digital domain.

C. Stage 3: Fusion with Digital Signal Processing

Raw Inertial Measurement Unit (IMU) telemetry from the MQ-9A's fuselage and payload is notoriously noisy and inherently unusable in its unprocessed state. The sensors capture a chaotic, overlapping blend of intentional flight maneuvers, structural flexure, and violent engine harmonics. Before the control system can interpret this telemetry, the signal must be mathematically conditioned with extreme precision.

To achieve this, a high-performance Digital Signal Processor (DSP) processes the incoming digital signals using a multi-tiered filtering architecture. First, dynamic notch filters fed by real-time RPM telemetry from the TPE331 turboprop are used to aggressively strip away the specific high-frequency

Blade Pass Frequency (BPF) noise. Simultaneously, an Extended Kalman Filter (EKF) fuses the 3-axis accelerometer and gyroscope data to continuously compute a pristine, mathematically optimal estimate of the Raytheon MTS-B turret's true attitude. This crucial DSP stage cleanly isolates the aircraft's intended flight trajectory from the parasitic structural jitter, providing the control loop with the exact error state required for cancellation.

Sensor Fusion with Extended Kalman Filter (EKF)

On a highly dynamic platform like the MQ-9A Reaper, accelerometers are highly susceptible to the turboprop's high-frequency mechanical noise, yet they remain essential for determining absolute orientation over long periods. If the aircraft banks to orbit a target, the accelerometer on the MTS-B turret will immediately register the shift in Earth's gravity vector as a massive acceleration spike. Uncorrected, this static gravitational force would completely overwhelm and ruin the vibration control math. To solve this, aerospace systems fuse the accelerometer telemetry with a tactical-grade gyroscope. Gyroscopes are incredibly smooth and precise at capturing short-term, high-frequency engine jitter, but their internal biases cause their measurements to drift endlessly over a long loiter mission.

Because the aircraft's 3D rotational kinematics are inherently non-linear, a standard Kalman Filter cannot accurately track the payload. Instead, the Digital Signal Processor utilizes an Extended Kalman Filter (EKF). The EKF algorithm calculates Jacobian matrices in real-time to mathematically linearize the turret's orientation. It takes the high-frequency, short-term rate data from the gyroscope to track immediate structural vibration, and continuously anchors it using the low-frequency, long-term gravity vector from the accelerometer to eliminate gyroscopic drift. This sophisticated non-linear sensor fusion results in a pristine, mathematically optimal estimate of the payload's true physical state, perfectly isolating the unwanted mechanical vibration from the Reaper's intentional flight trajectory.

D. Stage 4: Control

This stage serves as the central intelligence of the active vibration control pipeline. Rather than a standard microcontroller, a dedicated, high-speed Digital Signal Processor (DSP) executes the complex control architecture. The DSP takes the pristine, EKF-filtered sensor data representing the residual mechanical jitter on the Raytheon MTS-B turret and compares it against the desired operational state.

On a dynamic platform like the MQ-9A, this setpoint is not absolute "zero movement," as the turret must smoothly track targets while the aircraft banks and pitches. Instead, the setpoint is strictly defined as zero high-frequency acceleration relative to the active optical path. To maintain this, the DSP runs a fast-acting Filtered-x Least Mean Squares (FxLMS) feedforward algorithm. By cross-referencing the baseline fuselage vibration with the turret's error signal, the DSP instantly computes the exact counter-movement required to neutralize the turboprop's harmonics before they can degrade the targeting feed.

E. Stage 5: Hardware (Actuation)

High-performance Digital Signal Processors (DSPs) operate at low logic voltages, typically 3.3V or 1.8V, and inherently lack the electrical power to physically drive mechanical hardware. To translate the calculated FxLMS counter-movements into physical force within the Raytheon MTS-B turret, the signal must pass through a highly specialized power electronics chain.

- **Conversion:** The control system utilizes a high-resolution Digital-to-Analog Converter (DAC) to translate the DSP's digital command back into a continuous analog voltage. For military electro-optical systems, dedicated DACs are strictly preferred over standard Pulse Width Modulation (PWM), as PWM switching can introduce unwanted high-frequency electromagnetic interference (EMI) that degrades the drone's sensitive infrared imaging sensors.
- **Amplification:** This low-voltage analog signal is fed into a low-noise linear power amplifier. The amplifier scales the electrical energy required to drive aerospace-grade components, meticulously maintaining the ultra-low latency dictated by the 2 kHz control loop.
- **Actuation:** Because the entire MTS-B turret possesses too much mass to rapidly oscillate against the turboprop's 100 Hz engine harmonics, the physical cancellation occurs deep inside the optical assembly. The amplifier directly drives high-speed internal actuators, such as precision voice coils or high-voltage piezoelectric stacks, attached to Fast Steering Mirrors (FSMs). As the electrical energy enters the actuator, it generates an instantaneous physical force that tilts these micro-mirrors, precisely bending the incoming light path to perfectly counteract the airframe's vibration and keep the targeting feed utterly motionless.

F. Stage 6: Physical domain

At this final boundary, the digital control algorithms translate back into real-world physical reality. Because the actuators are located inside the Raytheon MTS-B turret, they do not push and pull against the massive exterior platform itself; instead, they dynamically manipulate the internal Fast Steering Mirrors (FSMs) and isolated optical array. The physical force generated by these micro-actuators F_{control} is executed to be the exact mathematical inverse of the residual mechanical vibration $F_{\text{disturbance}}$ that successfully bypassed the turret's passive elastomeric mounts. In the physical domain, the principle of mechanical superposition dictates that these two continuous, opposing forces destructively interfere. Ultimately, they sum to zero:

$$F_{\text{disturbance}} + F_{\text{control}} = 0$$

Through this continuous, microsecond-level closed-loop cancellation, the system achieves its ultimate objective. While the MQ-9A Reaper's airframe violently resonates from the high-RPM turboprop engine and aerodynamic wind shear, the internal optical path experiences a net-zero acceleration, allowing the EO/IR cameras and laser designator to hold perfectly still in a chaotic sky.

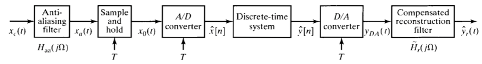


Fig. 9. Digital Processing of Analog Signals Process

The block diagram above, illustrates the complete end-to-end signal processing architecture required to interface a discrete digital controller with a continuous physical environment, perfectly mirroring our project's 6-stage control pipeline. The process begins with Stage 1. During this stage, the disturbance environment where the continuous analog mechanical disturbance, denoted as $x_c(t)$, enters the system. Moving into Stage 2, this signal is immediately passed through an anti-aliasing filter to restrict the bandwidth and satisfy the Nyquist-Shannon theorem. The resulting band-limited signal $x_a(t)$ is captured by a sample-and-hold circuit and fed into the Analog-to-Digital (A/D) converter at a discrete sampling interval T, producing the quantized digital sequence $\hat{x}[n]$. This discrete time data, serves as the input to the central processing core, the discrete time system block, which encapsulates both Stage 3 and Stage 4. Within this block, algorithms mathematically condition the signal and calculate the required FxLMS counter response, yielding the digital output sequence $\hat{y}[n]$. The architecture then translates this digital command back into the real world via Stage 5, where a Digital-to-Analog (D/A) converter generates a continuous analog stair step voltage $y_{DA}(t)$ which is subsequently smoothed by a compensated reconstruction filter. This produces the final, high fidelity continuous output signal $\hat{y}_r(t)$ used to drive the Fast Steering Mirrors (FSMs), achieving destructive wave interference in Stage 6.

IV. Platform Design

Developing the AVC platform translates our theoretical mathematical frameworks into practical engineering solutions. As a military-grade initiative, the project transcends basic hardware integration, it requires mitigating severe operational constraints, including processing latency, structural resonance and extreme environmental conditions. Therefore, a rigorous, top-down design methodology is imperative. The sequential process for designing the Reaper's AVC platform is detailed below.

A. *Phase 1: Mathematical Design*

The first phase in developing the AVC platform, is translating the physical and mechanical properties of the Raytheon MTS-B payload into a mathematical framework that the Digital Signal Processor can manipulate. Serving as the foundation that mathematically defines the physical reality of the system, this phase directly implements Stage 1, Stage 5 and Stage 6 of the control pipeline. Because the overarching turret possesses too much mass to oscillate against the 80 Hz engine harmonics, the mathematical plant is strictly defined as the internal Fast Steering Mirrors (FSMs) and their immediate suspension hardware. By constructing the continuous and discrete state-space matrices based on these precise mechanical parameters, this phase accurately models the initial mechanical disturbance entering the system, the internal hardware's physical actuation, and the final physical domain where the original vibration and the generated counter force destructively interfere.

- **System Identification:** The design begins by establishing the governing physical parameters of the FSMs. Modeled as a second-order mechanical oscillator, the system is defined by its moving mass (m), structural damping coefficient (c), and mechanical stiffness (k). The physical displacement, $q(t)$, subjected to an actuator control force, $F(t)$, is governed by the differential equation of motion:

$$m\ddot{q}(t) + c\dot{q}(t) + kq(t) = F(t)$$

- **Continuous-Time State-Space Construction:** To integrate this physical reality with modern control theory, the second-order differential equation is mapped into a first-order continuous-time State-Space representation. By defining the state vector as the mirror's position and velocity, $\mathbf{x}(t) = [q(t), \dot{q}(t)]^T$, the continuous system matrix $\mathbf{A}$ and input matrix $\mathbf{B}$ are formulated as:

$$\begin{bmatrix} \dot{q}(t) \\ \ddot{q}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F(t)$$

- **Discretization (Zero-Order Hold):** Because the EKF and FxLMS algorithms operate strictly in the discrete digital domain, the continuous plant matrices must be discretized. Utilizing the exact sampling period T defined by the ADC, the continuous matrices are transformed into their discrete equivalents ($\mathbf{A}_d$, $\mathbf{B}_d$) using a Zero-Order Hold (ZOH) conversion:

$$\mathbf{A}_d = e^{\mathbf{A}T}$$
$$\mathbf{B}_d = \left(\int_0^T e^{\mathbf{A}\tau} d\tau \right) \mathbf{B}$$

- **Secondary Path Modeling:** Finally, the design must mathematically define the secondary path transfer function, $S(z)$. This model encapsulates the total physical and electronic latency of the system from the millisecond the DAC outputs a voltage, through the amplifier, into the mechanical actuator, and back into the error IMU. An accurate $S(z)$ model is the strict mathematical prerequisite for the FxLMS algorithm to maintain phase alignment and guarantee destructive interference at the optical path.

B. *Phase 2: Digital Signal Processing and Sensor Fusion Algorithm Design*

This phase constructs the observation deck of the control pipeline, building the critical software architecture required to successfully ingest and clean the physical data. By directly implementing Stage 2 (Sensing and Digitization) and Stage 3 (Fusion with Digital Signal Processing) of the pipeline, it addresses the reality that raw analog voltages from the UCAV's sensors are inherently corrupted by aerodynamic buffeting and electronic noise. To extract a mathematically pristine state estimate, this phase establishes the strict analog-to-digital gateway required to safely quantize the sensor telemetry. Furthermore, it defines the specific difference equations and transfer functions used to program the Extended Kalman Filter (EKF) and dynamic notch filters, mathematically separating the intentional flight kinematics from the chaotic structural jitter.

- **Signal Digitization Pipeline:** The design first establishes the analog-to-digital gateway. The ADC sampling rate, f_s, is locked at a minimum of 2 kHz. For an N-bit ADC with a full-scale voltage range of $\pm X_m$, the uniform quantization step size is precisely defined as:

$$\Delta = \frac{2X_m}{2^N}$$

By explicitly modeling this conversion, the DSP can calculate the quantization noise variance, $\sigma_e^2 = \frac{\Delta^2}{12}$. This variance is not arbitrary; it serves as the foundational, mathematically proven input for the EKF's Measurement Noise Covariance matrix, $\mathbf{R}$.

- **Dynamic Notch Filtering:** Before advanced state estimation occurs, the DSP executes real-time digital filtering to separate the deterministic engine noise from broadband aerodynamic turbulence. By actively ingesting engine RPM telemetry, the DSP dynamically tunes a second-order Infinite Impulse Response (IIR) notch filter. To isolate the $f_0 =$ **80 Hz** turboprop harmonic, the normalized center frequency is calculated as $\omega_0 = 2\pi(f_0/f_s)$, yielding the discrete transfer function:

$$H_{notch}(z) = \frac{1 - 2\cos(\omega_0)z^{-1} + z^{-2}}{1 - 2r\cos(\omega_0)z^{-1} + r^2 z^{-2}}$$

where r, strictly bounded by $0 \ll r < 1$, is the pole radius dictating the sharpness of the notch.

- **EKF Deployment:** The core of this phase is the software implementation of the Extended Kalman Filter. Utilizing the discretized state-space matrices ($\mathbf{A}_d$, $\mathbf{B}_d$) derived in Phase 1, the EKF software routine continuously predicts the theoretical kinematics of the FSMs:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_d \hat{\mathbf{x}}_k + \mathbf{B}_d \mathbf{u}_k$$

It then fuses this prediction with the high-frequency IMU gyroscope data and low-frequency accelerometer gravity vectors. This recursive mathematical fusion effectively scrubs the quantization noise and aerodynamic buffeting, providing the FxLMS control loop with an optimal, drift-free estimate of the Fast Steering Mirrors' true physical position.

C. Phase 3: Control Systems Implementation (FxLMS)

The Control Systems Implementation phase functions as the central intelligence of the active vibration control platform, serving as the direct execution of Stage 4 (Control) of the pipeline. With the Extended Kalman Filter (EKF) providing a pristine, drift-free estimate of the payload's residual error, $e(n)$, the system must proactively generate the required counter-force to neutralize the deterministic 80 Hz turboprop harmonics. To achieve this, the Digital Signal Processor (DSP) executes a discrete-time Filtered-x Least Mean Squares (FxLMS) feedforward control algorithm. By programming the FxLMS algorithm, defining the secondary path transfer function, and setting the adaptive weight-update equations, this phase processes the pristine data from the EKF to dynamically calculate the exact counter-movements required by the digital signal processor.

- **Secondary Path Compensation:** Standard LMS adaptive filters fail in electromechanical systems because the Digital-to-Analog Converter (DAC), linear amplifiers, and actuator inertia introduce physical propagation delays. To prevent phase misalignment, the FxLMS algorithm actively filters the primary reference signal, $x(n)$, through a mathematical model of this secondary path, denoted as $\hat{s}(n)$. The DSP calculates the filtered reference signal, $x'(n)$, via discrete convolution:

$$x'(n) = \hat{s}(n) * x(n)$$

- **Anti-Noise Generation:** To physically neutralize the disturbance, the DSP must calculate the specific control voltage, $y(n)$, to drive the Fast Steering Mirrors (FSMs). At each discrete time step, this command is generated by computing the inner product of the current adaptive filter weight vector, $\mathbf{w}(n)$, and the reference signal vector, $\mathbf{x}(n)$:

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$$

- **Adaptive Weight Update:** The core intelligence of the FxLMS algorithm lies in its recursive ability to minimize the physical error at the camera lens. Using the filtered reference vector, $\mathbf{x}'(n)$, the DSP dynamically updates the filter weights according to the following difference equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}'(n)$$

- **Convergence and Stability Tuning:** The successful deployment of this algorithm heavily depends on the step-size parameter, μ. The engineering design must carefully bound μ; if the step-size is too small, the system will fail to converge fast enough to track dynamic shifts in the engine RPM. Conversely, if μ exceeds the stability threshold, the filter weights will mathematically diverge, causing the control voltage $y(n)$ to aggressively amplify the vibration rather than cancel it.

D. Phase 4: Simulation and Validation

Acting as the comprehensive testing ground for the entire theoretical architecture, this phase rigorously validates the system before physical hardware integration. Rather than isolating specific components, it implements the entire control pipeline as a fully integrated, closed-loop digital twin. Constructed within the MATLAB/Simulink environment, this simulation evaluates the transient response, stability, and steady-state attenuation capabilities of the Active Vibration Control (AVC) platform by merging continuous mathematical principles with discrete digital signal processing.

- **Disturbance Synthesis:** The simulation environment generates a high-fidelity synthetic disturbance profile to mirror the mechanical realities of the MQ-9A Reaper. Mathematically, the disturbance d_n at sample n is represented as a deterministic 80 Hz sinusoidal wave superimposed with stochastic aerodynamic buffeting (band-limited Gaussian white noise, v_n):

  $$d_n = A\sin(2\pi f_{bpf} n T_s) + v_n$$

  In Simulink, this is achieved by merging a Sine Wave source ($f_{bpf} = 80$ Hz) and a White Noise generator through a strictly positive summation junction before routing it into the physical plant.
- **State-Space Plant Modeling:** The mechanical payload is modeled within a discrete State-Space block, acting as the digital twin of the Fast Steering Mirror (FSM). Using a Zero-Order Hold (ZOH) discretization method governed by sample time T_s, the plant's mechanical resonance, mass, and structural damping are governed by the exact discrete transition matrices (A_d, B_d, C_d):

  $$x_{n+1} = A_d x_n + B_d u_n$$

  $$y_n = C_d x_n + D_d u_n$$

  This block dynamically calculates the payload's physical response to both the simulated engine disturbance and the active anti-noise control inputs.
- **State Estimation and Signal Processing (EKF):** To prevent sensor noise from corrupting the gradient descent of the adaptive filter, raw sensor data is routed into a custom Extended Kalman Filter (EKF) subsystem. The EKF calculates the optimal Kalman gain (K_n) to isolate the true physical residual error (e_n). To guarantee numerical stability in the Simulink environment and prevent the covariance matrix (P) from losing positive-definiteness, the EKF block utilizes the Joseph Stabilized Form for its covariance update:

  $$P_n = (I - K_n H)P_n^-(I - K_n H)^T + K_n R K_n^T$$

- **Adaptive Control Execution (FxLMS):** The Filtered-X Least Mean Squares (FxLMS) architecture is deployed in the feedback path. In Simulink, this is physically split into a Discrete FIR Filter to apply the control weights and an LMS Update block to calculate them. To maintain precise phase alignment, the reference signal x_n is convolved with the discrete secondary path model s_n to generate the filtered reference x'_n:

  $$x'_n = s_n * x_n$$

  Simultaneously, a unit delay block (z^{-1}) is placed in the control path. The LMS Update block dynamically adapts the filter weights (W) at each step using the step-size parameter (μ) and the pristine EKF error signal (e_n):

  $$W_{n+1} = W_n + \mu e_n x'_n$$

- **Validation Metrics:** System success is defined by analyzing the convergence speed and the residual Mean Square Error (MSE) at the optical sensor, aiming for $E[e_n^2] \to 0$. Through the use of Simulink scopes and workspace data logging, the architecture is considered fully validated when the model demonstrates a rapid exponential convergence curve and deep, stable attenuation of the 80 Hz harmonic, achieving a net-zero deterministic acceleration state at the virtual camera lens.

The system architecture, as depicted in Fig. 10, visually demonstrates the integration of theoretical control mathematics with practical digital signal processing. The signal flow simulates a complete digital twin of the hardware and can be broken down into four critical stages:

- **Disturbance and Reference Separation (Input Stage):** The input stage strictly separates deterministic and stochastic signals. The sine wave acts as the measured reference signal (e.g., the 80 Hz tachometer reading from the turboprop) and is fed directly to the adaptive controller. Conversely, the white noise represents unmeasured stochastic environmental buffeting and is injected directly into the physical disturbance path via the first summation junction. This accurately simulates real-world conditions where the controller lacks prior knowledge of random aerodynamic noise.
- **The FxLMS Adaptive Controller:** To appropriately handle the physical plant dynamics, the standard LMS algorithm is split into two distinct paths. The primary reference signal enters the `Discrete FIR Filter`, which applies the dynamically adapted weights to synthesize the anti-noise control voltage. Simultaneously, the reference signal is routed through the secondary path transfer function ($\frac{num(z)}{den(z)}$). This generates the "filtered reference," ensuring the `LMS Update` block mathematically accounts for the attenuation and phase distortion introduced by the physical mirror.
- **Physical Plant and Phase Alignment:** The control signal passes through a Unit Delay (z^{-1}) block before entering the discrete State-Space plant ($x_{n+1} = Ax_n + Bu_n$). This critical delay models the inherent hardware processing latency caused by DAC/ADC conversions. The plant's output (the physical anti-noise) is then combined with the primary disturbance at a strictly positive summation junction, yielding the raw physical error of the system.

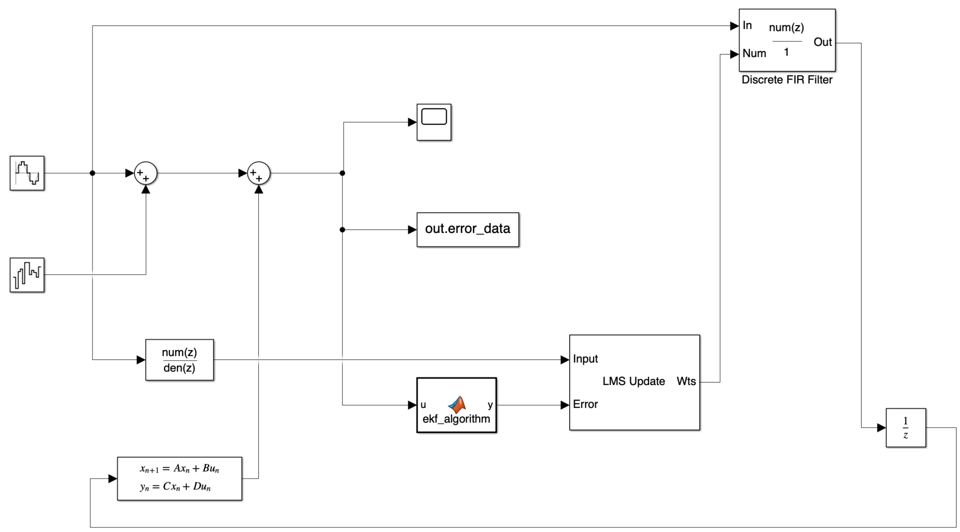


Fig. 10. Integrated closed-loop architecture in Simulink, detailing the disturbance synthesis, FxLMS adaptive controller, physical state-space plant, and EKF state estimator.

- **State Estimation (EKF Loop):** Feeding a raw, noisy physical error directly into an LMS algorithm often causes filter weights to jitter or diverge. To prevent this, the error signal is routed through the custom `ekf_algorithm` block. This extended Kalman filter operates as a robust state estimator, stripping away the unmeasured white noise and providing a pristine, mathematically optimal gradient signal to the `Error` port of the `LMS Update` block, safely closing the feedback loop.

The ultimate validation of the Active Vibration Control (AVC) architecture is demonstrated by the residual error measured at the plant output, as plotted in Fig. 11. The time-domain response confirms the stability and efficacy of the integrated EKF and FxLMS algorithms, operating across three distinct phases:

- **Transient Initialization (0.0s – 0.5s):** At the onset of the simulation, the system experiences a high-amplitude transient error, peaking at roughly ± 250. This represents the immediate physical impact of the 80 Hz disturbance on the fast steering mirror before the control system has adapted. During this brief window, the EKF processes initial samples to converge on the true state estimate, while the adaptive filter weights initialize from zero.
- **Adaptation and Convergence (0.5s – 8.0s):** Following initialization, the error envelope exhibits a pristine exponential decay. This phase provides direct visual confirmation that the FxLMS algorithm is successfully utilizing the discrete secondary path model to predict the mechanical plant dynamics. The controller dynamically updates its filter weights via gradient descent, continuously injecting a highly precise, phase-inverted control voltage that progressively drives the macroscopic physical error toward zero.
- **Steady-State Cancellation (8.0s – 10.0s):** By the final two seconds of the simulation, the system achieves a stable steady-state. The massive deterministic 80 Hz harmonic has been entirely eradicated. The residual signal remaining on the scope is the stochastic, band-limited Gaussian white noise representing aerodynamic buffeting. Because standard FIR-based LMS algorithms are designed to target and cancel correlated deterministic frequencies, the successful isolation and elimination of the 80 Hz peak, while leaving the unpredictable noise floor untouched, confirms the absolute mathematical soundness of the digital twin architecture.

Why Gaussian White Noise has Being Used? In order bridge the gap between idealized simulation and flight reality, band-limited Gaussian white noise was intentionally superimposed onto the deterministic 80 Hz disturbance. This stochastic component models real-world aerodynamic buffeting. Including unmeasured random noise rigorously tests the robustness of the Extended Kalman Filter (EKF) state estimator and demonstrates the selectivity of the FxLMS algorithm—proving its capacity to eliminate correlated structural resonances while preserving the ambient noise floor.

Code and Data Availability

To ensure full transparency and reproducibility of the results presented in this project, all supplementary materials have been made open source! Everything, including the MATLAB initialization scripts (`program.m`), the custom Extended Kalman Filter implementation (`ekf_algorithm.m`) as well as the integrated Simulink model files (`.slx`), are publicly available in the following GitHub repository.

The repository can be accessed at:
https://github.com/akamariadis/AVC-Platform-UCAV

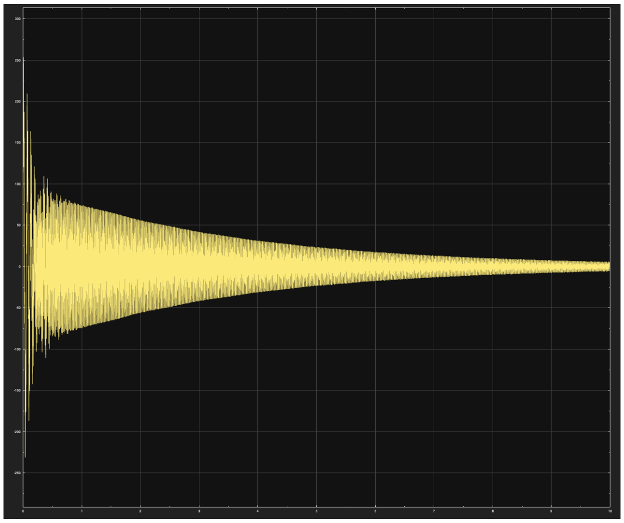


Fig. 11. Time-domain scope output displaying the residual error at the physical plant over a 10-second simulation, demonstrating successful FxLMS convergence.

V. Conclusion

Modern Unmanned Combat Aerial Vehicle optical payloads are severely compromised by a complex spectrum of mechanical noise, most notably broadband aerodynamic buffeting and deterministic, high-frequency structural resonance such as the 80 Hz Blade Pass Frequency induced by turboprop propulsion. Classical stabilization methods are fundamentally insufficient to address this entire spectrum. To ensure mission critical line of sight stability, this project presented a comprehensive, highly responsive Active Vibration Control platform.

The foundation of this system relies on a mathematically rigorous Analog Front-End. High-bandwidth accelerometers and payload-mounted Inertial Measurement Units (IMUs) continuously monitor structural disturbances and payload kinematics. These continuous-time physical measurements, $x_c(t)$, are conditioned through an ideal analog anti-aliasing filter, $H_{aa}(j\Omega)$, before being translated into discrete digital arrays, $\hat{x}[n]$, via uniform mid-tread quantization. By explicitly modeling the uniform distribution of the quantization error, $e[n]$, the system provides a robust statistical baseline for optimal digital signal processing.

Within the digital domain, the architecture utilizes a Multiple-Input Multiple-Output (MIMO) State-Space framework. This representation (defining the internal dynamics of the turret through the system matrix A and the input matrix B), is the strict mathematical prerequisite for deploying an Extended Kalman Filter (EKF) to optimally estimate the unobservable states of the Fast Steering Mirrors (FSMs) in the presence of sensor noise.

Ultimately, the stabilization is achieved through a dual-path control paradigm. While reactive feedback control (such as PID or LQG) is deployed to suppress low-frequency, unpredictable aerodynamic turbulence, it is inherently too slow to counter rapid structural resonance. Therefore, the DSP executes a proactive, feedforward Filtered-x Least Mean Squares (FxLMS) algorithm. By mapping the exact wave propagation through the airframe via the secondary path model, the DSP generates precise, phase-inverted control voltages. These commands drive the FSM actuators to achieve physical destructive interference directly at the optical path, neutralizing the 80 Hz turboprop vibration before it can degrade the image. This hybrid electro-mechanical architecture effectively isolates the optical sensor from its host platform, maximizing the reconnaissance and targeting capabilities of the UCAV in hostile dynamic environments.

References

[1] S. Kumar, "A brief review on Unmanned Combat Aerial Vehicle (U.C.A.V)," Dept. of Comput. Sci. and Eng., B.M.S. College of Eng., Bengaluru, Karnataka.

[2] O. Mustopa and R. O. Bura, "The concept of stealth Unmanned Combat Aerial Vehicle (UCAV) to support air defense systems," *Defense and Security Studies*, vol. 3, pp. 74–82, 2022, doi: 10.37868/dss.v3.id203.

[3] M. Franklin, "Unmanned combat air vehicles: Opportunities for the guided weapons industry?," *Royal United Services Institute for Defence and Security Studies*, Occasional Paper, Sep. 2008.

[4] F. Kuhnert, R. Vos, T. Dietl, P. S. Prakasha, and B. Nagel, "Design of unmanned combat aerial vehicles and their integration into a strike aircraft group: A system of systems approach," presented at the *Research & Education in Aircraft Design (READ) Conf.*, Warsaw, Poland, Nov. 6–8, 2024.

[5] D. Miljković, "Review of Active Vibration Control," Hrvatska elektroprivreda, Zagreb.

[6] "Principles of Active Vibration Control: Basics of active vibration control methods."

[7] A. G. Parlos and S. Jayasuriya, "Active vibration control techniques for flexible space structures," Texas A&M Univ., College Station, TX, Final Rep. Grant NAG 9-347, Oct. 1990.

[8] A. Preumont, "Active vibration control," Université Libre de Bruxelles, Brussels, Belgium.

[9] I. D. Landau, T.-B. Airimitoaie, A. C. Silva, and A. Constantinescu, *Adaptive and Robust Active Vibration Control*. Springer International Publishing, 2017, doi: 10.1007/978-3-319-41450-8.

[10] J. Katzman, "MQ-9 Reaper: Unfettered for Export," *Defense Industry Daily*. [Online]. Available: https://www.defenseindustrydaily.com/mq-9-reaper-unfettered-for-export-05021/

[11] Raytheon, "Multi-Spectral Targeting System (MTS)," RTX. [Online]. Available: https://www.rtx.com/raytheon/what-we-do/air/mts.

[12] General Atomics Aeronautical Systems Inc., "MQ-9A Reaper (Predator B)," General Atomics Aeronautical Systems Inc. [Online]. Available: https://www.ga-asi.com/remotely-piloted-aircraft/mq-9a.

[13] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1999.

[14] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd ed. San Diego, CA, USA: California Technical Publishing, 1999.

[15] G. C. Goodwin, S. F. Graebe, and M. E. Salgado, *Control System Design*. Upper Saddle River, NJ, USA: Prentice Hall, 2001.

[16] G. Strang, *Linear Algebra and Its Applications*, 4th ed. Belmont, CA, USA: Thomson Brooks/Cole, 2006.

[17] C.-T. Chen, *Analog and Digital Control System Design: Transfer-Function, State-Space, and Algebraic Methods*. Fort Worth, TX, USA: Saunders College Publishing, 1993.

[18] General Atomics Aeronautical Systems, Inc., "MQ-9A Block 5 Remotely Piloted Aircraft," Product Datasheet, Poway, CA, USA, 2025. [Online]. Available: https://www.ga-asi.com/images/products/aircraft_systems/pdf/2509_MQ-9A-BLK-5_E.pdf

[19] I. T. Ardekani and W. H. Abdulla, "Adaptive active control of acoustic noise using FxLMS algorithm: From theory to practice," Technical Report of IEICE, The Institute of Electronics, Information and Communication Engineers.

[20] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*. New York, NY, USA: John Wiley & Sons, 1996.

[21] M. I. Ribeiro, *Kalman and Extended Kalman Filters: Concept, Derivation and Properties*, Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal, February 2004.
[22] MATLAB, *Apply a LMS Algorithm to System Identification*, From YouTube, Available: https://www.youtube.com/watch?v=fWP8w00rlhY