sensors

MDPI

Article

Quantized State Estimation for Linear Dynamical Systems

Ramchander Rao Bhaskara [1,*], Manoranjan Majji [1] and Felipe Guzmán [2]

[1] Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843, USA; mmajji@tamu.edu
[2] Wyant College of Optical Sciences, The University of Arizona, Tucson, AZ 85721, USA; felipeguzman@arizona.edu
* Correspondence: bhaskara@tamu.edu

Abstract: This paper investigates state estimation methods for dynamical systems when model evaluations are performed on resource-constrained embedded systems with finite precision compute elements. Minimum mean square estimation algorithms are reformulated to incorporate finite-precision numerical errors in states, inputs, and measurements. Quantized versions of least squares batch estimation, sequential Kalman, and square-root filtering algorithms are proposed for fixed-point implementations. Numerical simulations are used to demonstrate performance improvements over standard filter formulations. Steady-state covariance analysis is employed to capture the performance trade-offs with numerical precision, providing insights into the best possible filter accuracy achievable for a given numerical representation. A low-latency fixed-point acceleration state estimation architecture for optomechanical sensing applications is realized on Field Programmable Gate Array System on Chip (FPGA-SoC) hardware. The hardware implementation results of the estimator are compared with double-precision MATLAB implementation, and the performance metrics are reported. Simulations and the experimental results underscore the significance of modeling quantization errors into state estimation pipelines for fixed-point embedded implementations.

Keywords: optical sensors; Kalman filter; state estimation; quantized filtering; finite-precision; FPGA

Citation: Bhaskara, R.R.; Majji, M.; Guzmán, F. Quantized State Estimation for Linear Dynamical Systems. *Sensors* **2024**, *24*, 6381. https://doi.org/10.3390/s24196381

Academic Editor: Chris Rizos

Received: 1 August 2024
Revised: 21 September 2024
Accepted: 29 September 2024
Published: 1 October 2024

1. Introduction

Kalman filters were first introduced into onboard guidance and navigation systems during NASA's Apollo Project in the 1960s [1]. However, digital simulations of the filter for onboard trajectory estimation using finite-precision arithmetic uncovered numerical stability problems. The filter simulations on the IBM 704 computer with 36-bit floating-point arithmetic were determined to be numerically unreliable. On the Apollo flight computer constrained with only 15-bit fixed-point arithmetic operations, onboard implementation of the filter was infeasible [2]. Since then, efforts to improve filter accuracy and stability under finite-precision hardware constraints have been a focus for practical realization of navigation filters. Square-root filtering was invented as a solution for execution on the Apollo guidance computer to solve the historic circumlunar navigation problem [3]. Square-root algorithms for state estimation were well-established by the 1970s [4,5] to address numerical stability issues due to quantization effects. With technological advancements in microprocessors and hardware accelerators, research on navigation filters continued to focus on high-speed, hard real-time calculations [6]. Recent research on quantized filtering algorithms has been enabled by optimized software packages, dedicated hardware for parallel computing, and purpose-built solutions [7–9].

The quantization problem in state estimation and optimal control has been widely studied in the literature [10–12]. Quantization is commonly utilized in wireless sensor networks (WSNs) with low-quality sensors with limited computation and communication capabilities, and where transmission bandwidth is severely constrained. A class of Kalman filters where the difference between measurements and its predictions (i.e., innovations)

are quantized has been developed for decentralized state estimation [12,13]. In these quantized Kalman filters, measurement predictions are distributed from a fusion center to a sensor node, where innovations are quantized and sent back to the center for filter update. The sign of innovation Kalman filter (SOI-KF) [13] operates by quantizing innovations to only 1-bit, which may give rise to large estimation errors. Filters are also proposed for multi-bit transmission, such as the multiple-level quantized innovation Kalman filter (MLQ-KF) [12]. It is shown that with multi-bit quantization, the performance of the quantized filter can be recovered to nearly match the mean squared errors of standard the Kalman filter [12]. However, to have such performance, uniformly distributed sensor nodes collectively convey their quantized information to a fusion center for processing into a combined target state estimate. Having just one isolated sensor significantly undermines estimator tracking performance, and a standard Kalman filter operating on high-resolution measurements from even one sensor is relatively more accurate. Work has also been reported in utilizing quantized measurements in the development and analysis of particle filters [14], unscented Kalman filters [15], and their fixed-point implementations [14,16]. These implementations are purpose-built for accelerating filter algorithms for specific applications and are not easily scalable for generic state estimation problems. To a great extent, the usual practice for high-speed onboard realization of filtering algorithms vastly depends upon the hardware resources and also on the choice of an optimal state estimator. This paper aims to bridge this gap by reformulating minimum mean square estimation algorithms to explicitly account for finite-precision numerical errors in states, inputs, and measurements. By integrating quantization noise models into the filter structure, the proposed approach enables the design of quantized filters that are better suited for fixed-point implementations on embedded systems.

This research focuses on optimal quantized filtering methods with an application focus on optomechanical acceleration sensors for inertial navigation. Optomechanical accelerometers rely on the coupling between the mechanical displacement of a test mass and light captured using an optical detection system [17]. Such a sensor requires optical data processing on high-speed hardware modules such as Field Programmable Gate Arrays (FPGAs) [18] for estimation of acceleration forces from test-mass displacement measurements. These precision force measuring units are deployed in geodesic applications, including the Gravity Recovery and Climate Experiment (GRACE) mission [19], the Laser Interferometer Space Antenna (LISA) [20], and the Laser Interferometer Gravitational-Wave Observatory (LIGO) [21].

For deployment on a spacecraft or resource-constrained hardware (e.g., FPGA [22,23]), dynamic range and sensor resolution limitations shall restrict the ability of standard filtering algorithms to precisely estimate states, especially when the dynamical system exhibits large variations in state variables. This necessitates the adoption of accurate error models and the use of novel filtering techniques to achieve sufficient estimation accuracy of desired state variables from the sensor measurements [10,12,24]. Furthermore, in scenarios where transmitting high-resolution measurements or performing double-precision operations is impossible or memory-intensive, quantization errors become significant. With increased performance requirements, the noise from the quantization effects has become an important aspect to analyze in efforts to maximize signal-to-noise ratio [12,13,25,26]. For the state estimation problem of dynamical stochastic processes, finite-precision implementation necessitates estimation to be based on quantized parameters of state, input, and observations. This requires revisiting the state estimation algorithms to include quantization errors in the filter implementation. Although using a sufficiently large word length for real-valued state variables can minimize the effects of quantization, even with a large number of bits, the system cannot be completely detached from finite word-length effects. In some instances, the luxury of large dynamic range for storage and computations may not be practicable. Consequently, the errors due to quantization have to be modeled as process and output noises, which can be accounted for in the system design using classical state-space and estimator modeling schemes.

The key contributions of this work are briefly summarized as follows. Given a finite-precision representation for system variables, linear state estimation filters are proposed. They consist of a (1) minimum variance estimator (least squares), quantized forms of (2) a discrete-time Kalman filter (QDKF), and (3) a square-root Kalman filter (QSRKF). The filter algorithms are applied to estimate states of an optomechanical oscillator model. The model formulation and assumptions are based upon the efforts of Kelly et al. [27] and the bench top experiments from Hines et al. [28]. Numerical simulations of the estimators are reported to show the prominence of modeling quantization errors in state estimation filters. The results offer much improved estimator performances over standard implementations of the respective filters when system variables are quantized. Steady-state filter performance is analyzed to provide insights into the best possible filter accuracy achievable for a given numerical representation. A least squares-based dual oscillator model is implemented on the FPGA board for the state estimation of acceleration forces from simulated measurements. A reliable estimator performance is reported by comparing the fixed-point implementation on the FPGA with the floating-point software implementation as a reference. Note that, in this work, quantization noise is from rounding off of numerical data to a desired number of bits. From here on, quantization errors and round-off errors are used interchangeably and refer to the same idea. Moreover, bit overflows are assumed to be negligible due to the careful selection of dynamic ranges for internal variable representation.

The rest of the article is organized as follows. The sensor model is described in Section 2. The quantization effects are incorporated into the model and state estimation filters are proposed in Section 3. Respective filter algorithms are derived in Appendices A–C. Numerical simulations are reported in Section 4 to illustrate the performance of the proposed filter structures. In Section 5, an FPGA architecture is proposed for on-board implementation of a dual-oscillator filter for acceleration estimation problem. Implementation results are reported. Concluding remarks are drawn in Section 6.

2. Description of Sensor Dynamics

The quantized filter performances are investigated for state estimation of the dynamical system described in this section. The generalized filtering algorithms are developed in Appendices A–C.

2.1. Discrete-Time Sensor Model

The one degree-of-freedom (1-DOF) accelerometer sensor dynamics are modeled as a second order spring-mass-damper system [29], equivalent to a perturbed linear harmonic oscillator. This formulation describes a direct conversion accelerometer, where the displacement of the proof mass $x(t)$ is directly measured using precise laser wavelength as a length reference [30]. Accounting for drift-causing optical and thermomechanical noise sources, through a white noise process and a biasing term, the dynamics for the proof mass displacement in response to the single-axis forcing function $g(t)$ are formulated using the following equations developed by Kelly [27] et al.:

$$\ddot{x} + 2\omega\zeta\dot{x} + \omega^2 x = g(t) + b(t) + n_v(t) \tag{1}$$

$$\dot{b}(t) = n_u(t) \tag{2}$$

where ω is the natural frequency of the oscillator, ζ is its damping factor, and the bias term $b(t)$ is modeled as a Wiener process. In discrete time, the increments of this Wiener process can be represented as an independent and identically distributed Gaussian random sequence. The terms $n_v(t)$ and $n_u(t)$ are uncorrelated, zero-mean Gaussian white-noise processes with spectral densities σ_v^2 and σ_u^2, respectively.

In continuous-time state space description, the 1-DOF accelerometer model is

$$\dot{\mathbf{X}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ -\omega^2 & -2\omega\zeta & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{X}(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} g(t) + \begin{bmatrix} 0 \\ n_v(t) \\ n_u(t) \end{bmatrix} \tag{3}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{X}(t) + \nu(t) \tag{4}$$

where the states of the system are $\mathbf{X}(t) = [x(t) \quad \dot{x}(t) \quad b(t)]^T$ representing displacement, velocity, and bias, in that order. The measurement model implicitly assumes that the position state, $x(t)$, is observable and that the sensor dynamics do not consider off-axis accelerations.

The linear system in Equation (3) can be discretized using a zero-order hold (ZOH) approximation, assuming that inputs and noise change only at discrete sampling intervals Δt. In this discretization, g_k is assumed constant over each sampling interval. However, it's crucial to note that if g_k varies more rapidly than can be observed within the interval Δt, the Nyquist theorem limits our ability to estimate g_k accurately. This highlights the importance of selecting an appropriate sampling rate in relation to the system's dynamics. The discretized state space model is

$$\mathbf{X}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_k)\mathbf{X}_k + \mathbf{\Gamma}(t_{k+1}, t_k)g_k + \mathbf{w}_k \tag{5}$$

where the transition from a state $\mathbf{X}_{k+1}$ to $\mathbf{X}_k$ for the linear time-invariant system is determined by a matrix exponential of the system matrix in Equation (3) (denoted as A):

$$\mathbf{\Phi}(t_{k+1}, t_k) = e^{A\Delta t} \tag{6}$$

Also,

$$\begin{aligned} \mathbf{\Psi}(t_{k+1}, t_k) &= \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau) d\tau \\ \mathbf{\Gamma}(t_{k+1}, t_k) &= \begin{bmatrix} \mathbf{\Psi}_{12}(t_{k+1}, t_k) & \mathbf{\Psi}_{22}(t_{k+1}, t_k) & \mathbf{\Psi}_{32}(t_{k+1}, t_k) \end{bmatrix}^T \end{aligned} \tag{7}$$

The stochastic process noise term $\mathbf{w}_k$, additive to the discrete-time state evolution, is described as

$$\begin{aligned} \mathbf{w}_k &= \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau) \begin{bmatrix} 0 & n_v(\tau) & n_u(\tau) \end{bmatrix}^T d\tau \\ \mathbf{Q} &= \mathbb{E}[\mathbf{w}_k \mathbf{w}_k^T] \end{aligned} \tag{8}$$

For the linear system described here, the state transition matrix $\mathbf{\Phi}$ and the covariance matrix $\mathbf{Q}$ associated with $\mathbf{w}_k$ can both be analytically derived [27]. $\mathbb{E}[\cdot]$ denotes the expected value operator.

Regarding the observation model, the displacement measurements are acquired at a fixed rate from an accurate one-dimensional interferometric displacement sensor. Following the state-space representation in Equation (4), the discrete-time measurement model is given by

$$y_k = x_k + \nu_k \tag{9}$$

This measurement model embeds the analog readout noise ν_k in the measurements. This observation error is treated as zero-mean white noise with an error variance of σ_m^2.

However, in practice, a change in the sensor's output does not always translate exactly to a change in the mechanical input. As a result, the one-to-one association between the measured and the physical displacements in Equation (27) may not always hold true. The deviation in the sensor's sensitivity is modeled through a scale factor and is estimated through sensor calibration and compensated during device operation [31]. The process of scale factor calibration and its associated uncertainties are not addressed in this paper.

However, the scale factor error arising from calibration uncertainty can be incorporated into the readout noise model for state estimation. To this extent, departure from unity scaling is modeled through scale factor error ϵ_s to account for the total readout error:

$$y_k = (1 + \epsilon_{s,k})x_k + \nu_k \tag{10}$$

Although not rigorously treated, this direction is briefly noted in Section 3.3.

2.2. Calibrated Sensor Model

Accelerometer bias accumulates over time, as indicated in Equation (2). The bias model is

$$b_{k+1} = b_k + \int_{t_k}^{t_{k+1}} n_u(\tau)d\tau \tag{11}$$

A reasonable calibration step must be implemented periodically to correct for the sensor bias to prevent its build-up effect on state estimation. A calibration sequence shall be performed by applying a known input to the sensor and observing the system response (as described in previous work [27]). This calibration step provides us with an estimate of the bias $\hat{b}$ and its error statistics as follows:

$$b_0 = \hat{b} + n_{b_0} \quad \text{and} \quad b_k = b_0 + \int_{t_0}^{t_k} n_u(\tau)d\tau \tag{12}$$

$$\mathbb{E}[b_k - \hat{b}] = 0 \tag{13}$$

$$\mathbb{E}[(b_k - \hat{b})(b_k - \hat{b})^T] = \sigma_{b0}^2 + (t_k - t_0)\sigma_u^2 \tag{14}$$

where, at the time of calibration t_0, the unbiased bias estimate b_0 is corrupted by a white noise source n_{b0} and has a variance of σ_{b0}^2. The bias estimate is assumed to not significantly degrade between periodic calibrations, but its variance grows as the bias evolves with a dependence upon the zero-mean Gaussian process $n_u(t)$ as shown in Equation (2).

Assuming that an independent calibration step has been performed to estimate the sensor bias, the model dynamics in position and velocity (Equation (5)), influenced by the instantaneous bias term b_k, can be written as follows:

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{\Phi}_{11} & \mathbf{\Phi}_{12} \\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_{22} \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \mathbf{\Psi}_{12} \\ \mathbf{\Psi}_{22} \end{bmatrix} g_k + \begin{bmatrix} \mathbf{\Phi}_{13} \\ \mathbf{\Phi}_{23} \end{bmatrix} \hat{b} + \tilde{\mathbf{w}}_k \\ \tilde{\mathbf{w}}_k &= \begin{bmatrix} \mathbf{\Phi}_{13} \\ \mathbf{\Phi}_{23} \end{bmatrix} (b_k - \hat{b}) + \int_{t_k}^{t_{k+1}} \begin{bmatrix} n_v(\tau)\mathbf{\Phi}_{12}(t_{k+1}, \tau) \\ n_v(\tau)\mathbf{\Phi}_{22}(t_{k+1}, \tau) \end{bmatrix} d\tau \end{aligned} \tag{15}$$

It can be shown that, to integrate the uncertainty in the instantaneous bias value into the system dynamics, the redefined process noise covariance $\tilde{\mathbf{Q}}$ is derived as:

$$\begin{aligned} \tilde{\mathbf{Q}} = &\begin{bmatrix} \mathbf{\Phi}_{13}^2 & \mathbf{\Phi}_{13}\mathbf{\Phi}_{23} \\ \mathbf{\Phi}_{23}\mathbf{\Phi}_{13} & \mathbf{\Phi}_{23}^2 \end{bmatrix} (\sigma_{b0}^2 + \sigma_u^2(t_k - t_0)) + \\ &\sigma_v^2 \begin{bmatrix} \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_{12}^2(t_{k+1}, \tau)d\tau & \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_{12}(t_{k+1}, \tau)\mathbf{\Phi}_{22}(t_{k+1}, \tau)d\tau \\ \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_{12}(t_{k+1}, \tau)\mathbf{\Phi}_{22}(t_{k+1}, \tau)d\tau & \int_{t_k}^{t_{k+1}} \mathbf{\Phi}_{22}^2(t_{k+1}, \tau)d\tau \end{bmatrix} \end{aligned} \tag{16}$$

The objective of this work is to estimate the forcing acceleration from position measurements. To achieve this, the governing equations for the system dynamics can be leveraged to estimate the states using either a minimum variance or a Kalman filter approach. However, the practical implementation of an onboard state estimation filter also accounts for additional artifacts arising from implementation on finite-precision computing architecture. In particular, discrete-time systems are susceptible to numerical errors when finite word-length registers are used to represent the states and measurements [25]. For accurate estimation of the forcing acceleration, it is essential to account for these finite word-length effects due to quantization. Neglecting such effects will degrade the estimation accuracy, as

will be shown through the numerical simulations that follow. Thus, quantization effects necessitate reformulating the state-space model described in Equation (5) to account for the corresponding numerical errors.

3. Model Reformulation and State Estimation

3.1. Dynamical System: Fixed-Point Realization

A fixed-point realization of a discrete-time system is a problem that considers the presence of quantization noise due to rounding off of products within the realization. Systematic approaches to deal with the adverse effects of fixed-point implementation in digital filters are developed by Mullis [26], Hwang [32], Williamson and Kadiman [33], Liu and Skelton [10]. These approaches involve formulating discrete-time models, such as those in Equations (5) and (9) where the system state $\mathbf{X}_k$, input g_k, and the measurement variable y_k are quantized at each instance of computation. The discrete-time model in Equation (5) is now redefined as

$$\mathbf{X}_{k+1} = \mathbf{\Phi}(t_{k+1}, t_k)\mathbb{Q}[\mathbf{X}_k] + \mathbf{\Gamma}(t_{k+1}, t_k)\mathbb{Q}[g_k] + \mathbf{w}_k \tag{17}$$

$$y_k = [1 \quad 0 \quad 0]\mathbb{Q}[\mathbf{X}_k] + \nu_k \tag{18}$$

where $\mathbb{Q}[.]$ here represents the quantization by round-off. The additive property of the round-off errors enables their modeling into the system description as

$$\mathbb{Q}[\mathbf{X}_k] = \mathbf{X}_k + \boldsymbol{\epsilon}_{\mathbf{x},k} \qquad \text{(state quantization)} \tag{19}$$

$$\mathbb{Q}[g_k] = g_k + \epsilon_{g,k} \qquad \text{(D/A conversion)} \tag{20}$$

$$\mathbb{Q}[y_k] = y_k + \epsilon_{y,k} \qquad \text{(A/D conversion)} \tag{21}$$

Here, $\boldsymbol{\epsilon}_{\mathbf{x},k}$ arises from quantization at the state nodes, $\epsilon_{g,k}$ is from digital-to-analog (D/A) conversion of input, and $\epsilon_{y,k}$ stems from rounding-off of the sampled measurements from an analog-to-digital (A/D) converter. This approach assumes that the state nodes are quantized after double-length accumulation. However, round-off errors in the coefficients are not independently treated in this model. It is assumed that coefficient round-off errors accumulate at the state nodes, and optimizing for state quantization tends to also account for coefficient quantization errors, as direct optimization of coefficient errors is not tractable [10].

Typically, round-off errors are characterized as zero-mean independent random variables that follow a uniform distribution [34]. This modeling approach accurately captures the inherent uncertainty associated with the precision of numerical computations. Therefore, the error statistics of round-off errors can be described as

$$\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},k}\} = \mathbf{0} \;\forall\; k \quad \text{and} \quad \mathbf{\Sigma}_{\mathbf{x}} = \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},k}\boldsymbol{\epsilon}_{\mathbf{x},k}^T\} = q_x I_{\mathbf{x}}; \quad q_x \triangleq \frac{2^{-2B_{\mathbf{x}}}}{12} \tag{22}$$

$$\mathbb{E}\{\epsilon_{g,k}\} = 0 \;\forall\; k \quad \text{and} \quad \Sigma_g = \mathbb{E}\{\epsilon_{g,k}\epsilon_{g,k}^T\} = q_g; \quad q_q \triangleq \frac{2^{-2B_g}}{12} \tag{23}$$

$$\mathbb{E}\{\epsilon_{y,k}\} = 0 \;\forall\; k \quad \text{and} \quad \Sigma_y = \mathbb{E}\{\epsilon_{y,k}\epsilon_{y,k}^T\} = q_y; \quad q_y \triangleq \frac{2^{-2B_y}}{12} \tag{24}$$

where $B_{\mathbf{x}}$, B_g, and B_y represent the word-lengths of state node registers and A/D and D/A converters. $I_{\mathbf{X}}$ is an identity matrix corresponding to the number of states. For a multi-output system where multiple measurements are available at an epoch, the measurement round-off covariance is $qI_{\mathbf{y}}$, as will be utilized in the subsequent derivation of the least squares estimator. By extension, for a multi-input system, the input round-off covariance could be defined as $qI_{\mathbf{g}}$. $I_{\mathbf{y}}$ and $I_{\mathbf{g}}$ have the dimensions that correspond to the number of inputs and outputs. The multi-input multi-output generalization is provided in Appendices A–C.

3.2. Formulation of Least Squares Estimator

The linear time-invariant system in Equation (15) implies that a measurement obtained at the nth instant from k ($n > k$) is related to the states and input at the instant t_k as

$$y_{k+n} = \begin{bmatrix} \mathbf{\Phi}_{11}^{(n)} & \mathbf{\Phi}_{12}^{(n)} & \mathbf{\Psi}_{12}^{(n)} \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \\ g_k \end{bmatrix} + \mathbf{\Phi}_{13}^{(n)} \hat{b} + \tilde{v}_{k+n} \tag{25}$$

$$\tilde{v}_{k+n} = v_{k+n} + \mathbf{\Phi}_{13}^{(n)} (b_k - \hat{b}) + \int_{t_k}^{t_{k+n}} \mathbf{\Phi}_{12}(t_{k+n}, \tau) n_v d\tau \tag{26}$$

where $\mathbf{\Phi}_{ij}^{(n)}$ and $\mathbf{\Psi}_{ij}^{(n)}$ represent the respective elements of $\mathbf{\Phi}(t_{k+n}, t_k)$ and $\mathbf{\Psi}(t_{k+n}, t_k)$ (refer Equation (5)).

For a batch of $N+1$ position measurements during which the input acceleration and the bias are assumed unchanged, the instantaneous states and the acceleration input are linearly related as follows:

$$\underbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+N} \end{bmatrix}}_{\tilde{\mathbf{y}}} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \mathbf{\Phi}_{11}^{(1)} & \mathbf{\Phi}_{12}^{(1)} & \mathbf{\Psi}_{12}^{(1)} \\ & \vdots & \\ \mathbf{\Phi}_{11}^{(N)} & \mathbf{\Phi}_{12}^{(N)} & \mathbf{\Psi}_{12}^{(N)} \end{bmatrix}}_{\mathbf{H}_k} \underbrace{\begin{bmatrix} x_k \\ \dot{x}_k \\ g_k \end{bmatrix}}_{\tilde{\mathbf{X}}_k} + \underbrace{\begin{bmatrix} 0 \\ \mathbf{\Phi}_{13}^{(1)} \\ \vdots \\ \mathbf{\Phi}_{13}^{(N)} \end{bmatrix}}_{\eta_k} \hat{b} + \underbrace{\begin{bmatrix} \tilde{v}_k \\ \tilde{v}_{k+1} \\ \vdots \\ \tilde{v}_{k+N} \end{bmatrix}}_{\tilde{\boldsymbol{v}}_k} \tag{27}$$

where the forcing acceleration input g_k is now treated as an additional state for estimation.

Additionally, incorporating round-off errors described in Equation (17) into the above model yields:

$$\tilde{\mathbf{y}} = \mathbf{H}_k(\tilde{\mathbf{X}}_k + \boldsymbol{\epsilon}_{\tilde{\mathbf{X}},k}) + \boldsymbol{\eta}_k(\hat{b} + \epsilon_{\hat{b},k}) + \tilde{\boldsymbol{v}}_k + \boldsymbol{\epsilon}_{\tilde{\mathbf{y}}} \tag{28}$$

where the bias round-off error ($\epsilon_{\hat{b},k}$) is assumed to have the same variance as that of state (q_x). The input round-off error from D/A, $\epsilon_{g,k}$, is absorbed into the estimation state error $\boldsymbol{\epsilon}_{\tilde{\mathbf{X}},k}$. Finally, the uncorrelated errors due to quantization in the measurement batch are accumulated in a vector of length $N+1$ as $\boldsymbol{\epsilon}_{\tilde{\mathbf{y}}}$.

Grouping the error terms together in a new variable $\boldsymbol{\mu}$, the above linear system of equations can be expressed as

$$\begin{aligned} \tilde{\mathbf{y}} - \boldsymbol{\eta}_k \hat{b} &= \mathbf{H}_k \tilde{\mathbf{X}}_k + \boldsymbol{\mu} \\ \boldsymbol{\mu} &= \mathbf{H}_k \boldsymbol{\epsilon}_{\tilde{\mathbf{X}},k} + \boldsymbol{\eta}_k \epsilon_{\hat{b},k} + \tilde{\boldsymbol{v}}_k + \boldsymbol{\epsilon}_{\tilde{\mathbf{y}}} \end{aligned} \tag{29}$$

The measurement error mean and the covariance ($\mathbf{P}_{\mu\mu}$) can be obtained as

$$\mathbb{E}[\boldsymbol{\mu}] = \mathbf{H}_k \mathbb{E}[\boldsymbol{\epsilon}_{\tilde{\mathbf{X}},k}] + \boldsymbol{\eta}_k \mathbb{E}[\epsilon_{\hat{b},k}] + \mathbb{E}[\tilde{\boldsymbol{v}}_k] + \mathbb{E}[\boldsymbol{\epsilon}_{\tilde{\mathbf{y}}}] = \mathbf{0} \tag{30}$$

$$\mathbf{P}_{\mu\mu} = \mathbb{E}[\boldsymbol{\mu}\boldsymbol{\mu}^T] = \mathbf{H}_k \mathbf{\Sigma}_{\tilde{\mathbf{X}}} \mathbf{H}_k^T + \boldsymbol{\eta}_k \Sigma_{\hat{b}} \boldsymbol{\eta}_k^T + \mathbf{P}_{\tilde{v}\tilde{v}} + \mathbf{\Sigma}_{\tilde{\mathbf{y}}} \tag{31}$$

where the quantization noise covariances ($\mathbf{\Sigma}_{\tilde{\mathbf{X}}}$, $\Sigma_{\hat{b}}$, and $\mathbf{\Sigma}_{\tilde{\mathbf{y}}}$) are obtained from Equations (22)–(24) as

$$\mathbf{\Sigma}_{\tilde{\mathbf{X}}} = \mathrm{diag}(q_x, q_x, q_g); \quad \Sigma_{\hat{b}} = q_x; \quad \text{and} \quad \mathbf{\Sigma}_{\tilde{\mathbf{y}}} = q_y \mathbf{I}_{\tilde{\mathbf{y}}} \tag{32}$$

Moreover, the elements of the measurement noise covariance matrix $\mathbf{P}_{vv}$ have explicit dependence on time and will require periodic calibration to prevent measurement degradation [27]. The covariance elements can be computed in indicial notation as

$$\mathbf{P}_{\nu\nu,ij} = \sigma_m^2\delta_{ij} + \mathbf{\Phi}_{13}^{(i)}\mathbf{\Phi}_{13}^{(j)}\left(\sigma_{b0}^2 + \sigma_u^2(t_k - t_0)\right) + \sigma_v^2\int_{t_k}^{t_{k+i}} \mathbf{\Phi}_{12}^2(t_{k+i},\tau)d\tau \tag{33}$$

Ultimately, an optimal state estimate using the batch of measurements $\tilde{\mathbf{y}}$, are obtained by solving the normal equations as

$$\hat{\tilde{\mathbf{X}}}(k) = [\mathbf{H}_k^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}_k]^{-1}\mathbf{H}_k^T\mathbf{P}_{\mu\mu}^{-1}(\tilde{\mathbf{y}} - \boldsymbol{\eta}_k\hat{b}) \tag{34}$$

The above expression is a stochastic moving average filter, and it consumes at least three running measurements ($N \geq 3$) to provide the estimates for the single-axis position, velocity, and acceleration states at every estimation epoch. Evidently, the filter averages the measurements from two future time events, causing the filter to lag the measurement sequence by at least two measurement cycles. The minimum variance estimation with quantized states and measurements for a linear system is derived in Appendix A.

3.3. Note on State Estimation with Scale Factor Errors

The readout error in Equation (26) can be modified to account for the scale factor error described in Equation (10). This modification starts with acknowledging that the measurements are contaminated by measurement noise and scale factor errors as

$$y_{k+n} = (1 + \epsilon_{s,k+n})x_{k+n} + \nu_{k+n} \tag{35}$$

wherein the scale factor error is modeled as an uncorrelated zero-mean Gaussian white noise with variance $\sigma_{\epsilon_s}^2$.

As a consequence, the total readout error in Equation (26) has an additional term $\epsilon_{s,k+n}x_{k+n}$ such that

$$\tilde{v}_{k+n} = \epsilon_{s,k+n}x_{k+n} + \nu_{k+n} + \mathbf{\Phi}_{13}^{(n)}(b_k - \hat{b}) + \int_{t_k}^{t_{k+n}} \mathbf{\Phi}_{12}(t_{k+n},\tau)n_v d\tau \tag{36}$$

Notice that the measurement noise is now linearly related to the displacement state. This problem can be addressed by using an *a priori* estimate of displacement state ($\hat{x}_{a,k+n}$) from the propagation of model dynamics (Equation (15)) [35].

Now, using $\hat{x}_{a,k+n}$ for evaluating the scale factor error contribution, it can be shown that the additional variance that contributes to the measurement noise variance is

$$\sigma_{\tilde{v}_{k+n}}^2 = \sigma_{\nu_{k+n}}^2 + \sigma_{\epsilon_{s,k+n}}^2(\hat{x}_{a,k+n})^2 \tag{37}$$

Following a procedure similar to the one described in Section 3.2, this updated measurement noise variance can now be used to compute $\mathbf{P}_{\tilde{v}\tilde{v}}$ in Equation (31). Note that the incorporation of *a priori* state information and corresponding *a priori* error covariance ($\tilde{\mathbf{Q}}$) allows us to extend the least squares filter to update the state estimates as described in Appendix A.3. Additionally, since acceleration is an estimated state, its contribution to the process noise is evaluated as a tunable parameter (details in Section 3.5).

3.4. Multi-Oscillator Problem

The dynamic response of a harmonic oscillator depends on the oscillator's natural frequency, damping, and the driving signal frequency. To attain a wide dynamic range for precise inertial measurements, multiple oscillators may be deployed in a multiplexed sensing network [28]. The estimates of forcing accelerations from multiple sensing nodes are appropriately fused to obtain an estimate with high confidence. If the estimates are calculated from the observations sampled by the same hardware system at the same instance, a covariance-weighted average of all the arriving estimates can be implemented

to reduce noise over a wide working range of measurable accelerations using the multi-oscillator system's response.

Assuming that the estimation errors (from least squares filter independently applied to measurements from different oscillators) are independent and unbiased (zero-mean), the acceleration estimates are fused by using a weighted average with the reciprocal variance values as weights [36]:

$$\overline{g}(k) = \frac{\sum_{i=0}^{i=N} \hat{g}_i(k) \cdot \frac{1}{\mathbf{P}_{33}^i}}{\sum_{i=0}^{i=N} \frac{1}{\mathbf{P}_{33}^i}} \tag{38}$$

where N represents the number of independent estimates and $\mathbf{P}_{33}^i$ is the estimated non-zero variance of $\hat{g}_i(k)$. The variance $\sigma_{\overline{g}}^2$ of the fused estimate is always lower or equal to the best individual estimate and is given by

$$\sigma_{\overline{g}}^2 = \sum_{i=0}^{i=N} \frac{1}{\mathbf{P}_{33}^i} \tag{39}$$

Note that the independence of error in estimates is a loosely constructed term and difficult to fulfill, especially when the oscillators are of the same type. If the individual estimation errors are not independent, the covariance-weighted average still yields a correct estimate, but its assigned confidence is overestimated.

3.5. Quantized Discrete-Time Kalman Filter (QDKF)

The least squares moving average filter described in Section 3.2 uses an $N+1$ measurement batch to estimate the states at every epoch. In this section, a Kalman filter formulation is briefly described where the state variables are sequentially estimated by fusing predictions of the state variables from the oscillator dynamical model with noisy position measurements.

The discrete-time dynamics given in Equation (15) can be remodeled to accommodate the instantaneous forcing acceleration input as a state. However, since the forcing acceleration is not directly observed through a model, it is estimated from new measurements. The process covariance is augmented with an acceleration model uncertainty parameter α that is appropriately scaled to indicate the confidence in the evolution of g_k. After all, the acceleration input cannot be perfectly delivered to the system and is affected by a noise process that is denoted here as $w_{g,k}$. An independent periodic calibration step prevents accumulation of errors in acceleration estimates. The dynamical model therefore can be reformulated with the evolution of the modified states, $\mathbf{X}_k = [x_k \quad \dot{x}_k \quad g_k]^T$, as given by the discrete-time dynamics as

$$\underbrace{\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \\ g_{k+1} \end{bmatrix}}_{\mathbf{X}_{k+1}} = \underbrace{\begin{bmatrix} \mathbf{\Phi}_{11} & \mathbf{\Phi}_{12} & \mathbf{\Psi}_{12} \\ \mathbf{\Phi}_{21} & \mathbf{\Phi}_{22} & \mathbf{\Psi}_{22} \\ 0 & 0 & 1 \end{bmatrix}}_{\widetilde{\mathbf{\Phi}}} \underbrace{\begin{bmatrix} x_k \\ \dot{x}_k \\ g_k \end{bmatrix}}_{\mathbf{X}_k} + \underbrace{\begin{bmatrix} \mathbf{\Phi}_{13} \\ \mathbf{\Phi}_{23} \\ 0 \end{bmatrix}}_{\widetilde{\mathbf{\Gamma}}} \hat{b} + \underbrace{\begin{bmatrix} \tilde{\mathbf{w}}_k \\ w_{g,k} \end{bmatrix}}_{\bar{\mathbf{w}}_k} \tag{40}$$

Finally, the dynamical and the measurement models, under the influence of quantization noises, can be reformulated as

$$\mathbf{X}_{k+1} = \widetilde{\mathbf{\Phi}}_k(\mathbf{X}_k + \boldsymbol{\epsilon}_{\mathbf{X},k}) + \widetilde{\mathbf{\Gamma}}_k(\hat{b} + \epsilon_{\hat{b},k}) + \bar{\mathbf{w}}_k \tag{41}$$

$$y_k = \mathbf{H}_k(\mathbf{X}_k + \boldsymbol{\epsilon}_{\mathbf{X},k}) + \tilde{\boldsymbol{\nu}}_k + \epsilon_y \tag{42}$$

where $\mathbf{H}_k = [1 \quad 0 \quad 0]$ is the observation model matrix indicating that the position of the proof mass is directly observable through measurements. The round-off errors in-

cluded in the model follow the same definitions as described in the least squares estimator (Section 3.2).

With the reformulated dynamics and the observation models described in Equations (41) and (42), the round-off errors are incorporated into the Kalman filter formulation for sequential state estimation. Starting from an initial value of state and corresponding error covariance, a quantized form of Kalman filter, QDKF, is thus realized. Algorithm 1 describes the recursive operations involved in the implementation of the QDKF. The derivation of the QDKF is presented in Appendix B.

Algorithm 1: Quantized discrete-time Kalman filter (QDKF).

1: Initialize

$$\hat{\mathbf{X}}_0^+ = \mathbb{E}[\mathbf{X}_0] \tag{43}$$

$$\mathbf{P}_0^+ = \mathbb{E}[(\hat{\mathbf{X}}_0 - \mathbf{X}_0)(\hat{\mathbf{X}}_0 - \mathbf{X}_0)^T] \tag{44}$$

2: Propagate

$$\hat{\mathbf{X}}_{k+1}^- = \widetilde{\mathbf{\Phi}}_k \hat{\mathbf{X}}_k^+ + \widetilde{\mathbf{\Gamma}}_k \hat{b} \tag{45}$$

$$\mathbf{P}_{k+1}^- = \widetilde{\mathbf{\Phi}}_k (\mathbf{P}_k^+ + \mathbf{\Sigma}_{\mathbf{X},k}) \widetilde{\mathbf{\Phi}}_k^T + \widetilde{\mathbf{\Gamma}}_k \mathbf{\Sigma}_{\hat{b},k} \widetilde{\mathbf{\Gamma}}_k^T + \begin{bmatrix} \tilde{\mathbf{Q}} & 0 \\ 0 & \alpha \end{bmatrix} \tag{46}$$

3: Update

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{H}_k \mathbf{\Sigma}_{\mathbf{X},k} \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{\Sigma}_{\mathbf{y},k}]^{-1} \tag{47}$$

$$\hat{\mathbf{X}}_k^+ = \hat{\mathbf{X}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{X}}_k^-) \tag{48}$$

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- \tag{49}$$

An equivalent expression for the covariance update in Equation (49) can be written in a symmetric form, as shown below. This symmetric version is often used in software implementation as it guarantees positive semi-definiteness of $\mathbf{P}_k^+$ in the presence of round-off errors.

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^- [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]^T + \mathbf{K}_k [\mathbf{H}_k \mathbf{\Sigma}_{\mathbf{X},k} \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{\Sigma}_{\mathbf{y},k}] \mathbf{K}_k^T \tag{50}$$

3.6. Quantized Square-Root Kalman Filter (QSRKF)

For onboard implementation with limited computational word-length, the standard Kalman filter algorithm is susceptible to numerical instability. Round-off errors can cause loss of positive definiteness in error covariance matrices during computation [2,37]. Square-root Kalman filters (SRKFs) mitigate this numerical degradation by computing and propagating the square roots of the error covariance matrices for both time and measurement updates. The quantized version of square-root filters, termed QSRKF here, incorporates quantization errors into the SRKF formulation, thereby improving performance as well as the filter's numerical stability under fixed-point implementation.

In QSRKF, error covariance matrices are factored into square-root matrices computed using QR decomposition. The square roots of the initial state error, process noise, measurement noise, and quantization noise covariance matrices are calculated once using the Cholesky method. For linear time-invariant systems, these can often be predetermined and stored onboard. For the dynamical system described in the QDKF formulation (Section 3.5), Algorithm 2 presents the operations involved in QSRKF realization. The filter algorithm is described in detail in Appendix C.

Algorithm 2: Quantized Square-Root Kalman filter (QSRKF).

1: Initialize

$$\hat{\mathbf{X}}_0^+ = \mathbb{E}[\mathbf{X}_0] \tag{51}$$

$$\mathbf{S}_0^+ = \sqrt{\mathbb{E}[(\hat{\mathbf{X}}_0 - \mathbf{X}_0)(\hat{\mathbf{X}}_0 - \mathbf{X}_0)^T]} \tag{52}$$

2: Propagate

$$\hat{\mathbf{X}}_{k+1}^- = \widetilde{\mathbf{\Phi}}_k \hat{\mathbf{X}}_k^+ + \widetilde{\mathbf{\Gamma}}_k \hat{b} \tag{53}$$

$$\mathbf{S}_{k+1}^- = \mathrm{qr}\{\left[\mathbf{\Phi}_k \mathbf{S}_k^+ \mid \mathbf{\Phi}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{\Gamma}_k \mathbf{\Lambda}_{\mathbf{u},k} \mid \gamma_k \mathbf{S}_{\mathbf{w},k}\right]^T\}^T \tag{54}$$

3: Update

$$\mathbf{S}_{\mathbf{zz},k} = \mathrm{qr}\{\left[\mathbf{H}_k \mathbf{S}_k^- \mid \mathbf{H}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{S}_{\mathbf{v},k} \mid \mathbf{\Lambda}_{\mathbf{y},k}\right]^T\}^T \tag{55}$$

$$\mathbf{K}_k = \mathbf{S}_k^- (\mathbf{H}_k \mathbf{S}_k^-)^T (\mathbf{S}_{\mathbf{zz},k} \mathbf{S}_{\mathbf{zz},k}^T)^{-1} \tag{56}$$

$$\hat{\mathbf{X}}_k^+ = \hat{\mathbf{X}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{X}}_k^-) \tag{57}$$

$$\mathbf{S}_k^+ = \mathrm{qr}\left\{\left[[\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]\mathbf{S}_k^- \mid \mathbf{K}_k \left[\mathbf{H}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{S}_{\mathbf{v},k} \mid \mathbf{\Lambda}_{\mathbf{y},k}\right]\right]^T\right\}^T \tag{58}$$

The following is true for the above algorithm:

- $\mathbf{S}_0^+$ is the square-root factor of the initial estimation error covariance.
- $\mathbf{S}_{k+1}^-$ and $\mathbf{S}_k^+$ represent the prior and posterior estimation error covariance square-root factors.
- $\mathbf{S}_{\mathbf{v},k}$ and $\mathbf{S}_{\mathbf{w},k}$ are the square-roots of the measurement and process noise covariance matrices.
- $\mathbf{\Lambda}_{\mathbf{x},k}$, $\mathbf{\Lambda}_{\mathbf{y},k}$, and $\mathbf{\Lambda}_{\mathbf{u},k}$ represent the matrix square-roots of the quantization error covariances for state, measurement, and output noises.
- $\mathrm{qr}\{\cdot\}$ indicates QR decomposition operation.

4. Numerical Simulations

The optomechanical inertial sensor parameters and the corresponding noise processes modeled in this simulation are highlighted in Table 1. These parameters are derived from laboratory experiments and sensor benchtop prototypes described in references [28,31,38]. The contributing process noise sources, including thermal, mechanical, and cavity drifts, have been thoroughly investigated in the cited works. The oscillator parameters, process noise terms (σ_v, σ_u) driving the discrete-time state evolution, calibrated sensor bias as well as the readout noise floor [27], are set to values consistent with the experimental observations.

Noisy position measurements are simulated from true position states corrupted by a stochastic measurement noise process (σ_m) along with an additive quantization process (q_y). The stochastic processes (σ_v, σ_u) driving the discrete-time state evolution, the state quantization processes (q_x), the standard deviation of the calibrated bias estimate (σ_{b0}), and its corresponding quantization process ($q_{\hat{b}}$) are also presented in Table 1.

In this simulation, sensor bias is assumed to be estimated through an independent calibration step that is briefly described in Section 2.2 and detailed in [27]. The bias estimate $\hat{b}$ is randomly drawn from $\mathcal{N}(0, \sigma_{b0}^2)$, and the acceleration input is simulated as a sinusoidal signal of frequency 0.01 Hz and amplitude 1×10^{-5} g. In this analysis, the state and bias values are stored with fractional word length of internal registers configured to signed 16 bits ($B_x = B_{b0} = 16$). This means that the fractional parts of the states and input nodes are rounded off to 16 bits, with the word-length of the measurements being the only variable. The measurements are quantized to fractional lengths (FLs) of 8 to 16 bits (B_y) to emulate different ADC resolutions available on development boards. To prevent numerical under-

flows and to maintain precision, the covariance matrix elements are assigned higher word lengths compared with the state and measurement variables. For FPGA implementation, dedicated digital signal processing (DSP) blocks can be utilized to efficiently multiply the covariance matrix elements with the corresponding state or measurement variables. Modern FPGAs typically provide DSP blocks that support multiplication of 18-bit and 25-bit operands. To handle higher word lengths of the covariance matrix elements, multiple DSP blocks can be cascaded to perform pipelined multiply-and-accumulate operations, enabling accurate computations while maintaining high throughput.

Table 1. Simulated sensor model and noise parameters.

Parameter	Value	Units
Oscillators		
Sampling frequency (f_s)	30.5	Hz
Oscillator-1 frequency (ω_1)	3.76	Hz
Oscillator-2 frequency (ω_2)	8.5	Hz
Damping ratios (ζ_1, ζ_2)	4.386×10^{-6}	
Modeling processes		
σ_v	1×10^{-9}	$\mathrm{m/s}\sqrt{\mathrm{Hz}}$
σ_u	1×10^{-8}	$\mathrm{m/s^2}\sqrt{\mathrm{Hz}}$
σ_m	1×10^{-11}	m
σ_{b0}	1×10^{-8}	$\mathrm{m/s^2}$
Quantization processes		
q_x	$\sqrt{2^{-2B_x}/12}$	unit of corresponding state
q_y	$\sqrt{2^{-2B_y}/12}$	m
$q_{\hat{b}}$	$\sqrt{2^{-2B_{\hat{b}}}/12}$	$\mathrm{m/s^2}$

Figure 1 shows the results for the errors in the estimated acceleration and the corresponding 3σ estimation error bounds from the moving average least squares method. In this figure, the errors and the 3σ bounds for the least squares implementation are compared for measurements of 12 versus 16-bit fractional length. The least squares filter structure defined in Equation (34) supplements the measurement noise covariance with the quantization error covariance that reflects the bit resolution. Hence, the simulation in Figure 1 demonstrates additional errors in the acceleration estimates due to less precise representation of measurements. The simulation also signifies the expansion of the 3σ bounds, indicative of added uncertainties in the estimated errors due to increased quantization noise in the measurements. For instance, acceleration estimates with the measurements rounded off to a fractional length of 12-bits have larger errors and covariance bounds than those of 16-bit measurements.

Figure 2 shows the estimation error results for the Kalman filter formulation, where the process noise associated with the acceleration channel is modeled as a zero-mean process with a variance of 0.1 ($\alpha = 0.1$). As in the least squares method, the errors and error bounds from the Kalman filter are sensitive to the precision in the measurements. This sensitivity is illustrated in Figure 2a,b, which respectively compare 8 and 12-bit measurements against measurements with 16-bit fractional resolution. The analysis suggests that as the measurement precision increases, the errors and the error bounds tend to align statistically with those of a floating-point implementation of the discrete-time Kalman filter, despite finite-precision hardware constraints. In another observation, the least squares filter has lower variance than that of the QDKF for this application. This is because the least squares filter need not account for the process noise associated with the acceleration channel.

Furthermore, accounting for finite word-length effects in the filter implementation not only accurately predicts the uncertainty in the estimation errors but also reduces these errors. Shown in Figure 3 is discrete-time Kalman filter (DKF) implementation without considering round-off errors in the filter design versus the quantized discrete-time Kalman filter (QDKF) proposed in this work (Algorithm 1). In this comparison, the measurements

are quantized to 12-bits in both the DKF and the QDKF implementations, but the filter formulation in DKF does not incorporate quantization error covariances as the proposed QDKF algorithm does. Evidently, the QDKF filter resulted in lower error values than the DKF and also higher 3σ bounds to represent increased covariance due to quantization errors. Moreover, the DKF errors are numerically inconsistent with the corresponding error covariances, as its formulation does not account for quantization noise statistics. The same phenomenon is also evident in the least squares estimation, as observed in Figure 1. Furthermore, numerical round-off errors exacerbate the lack of observability in the bias state, resulting in growing uncertainty in acceleration estimation errors.

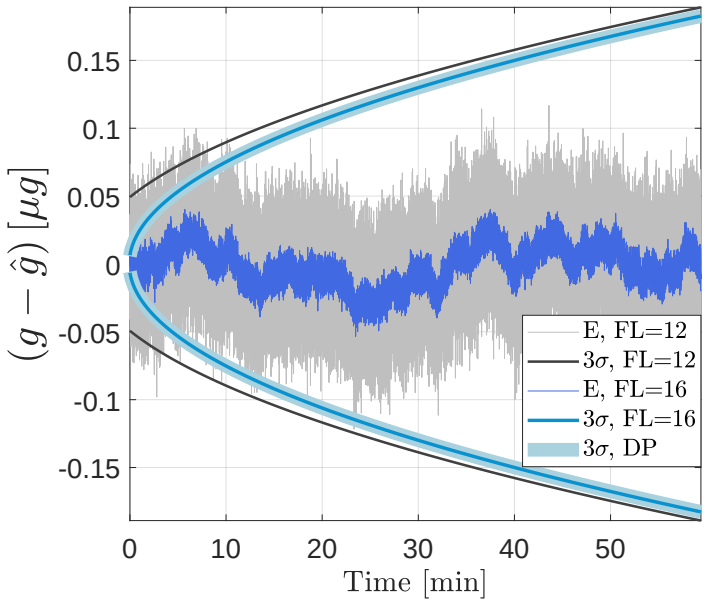
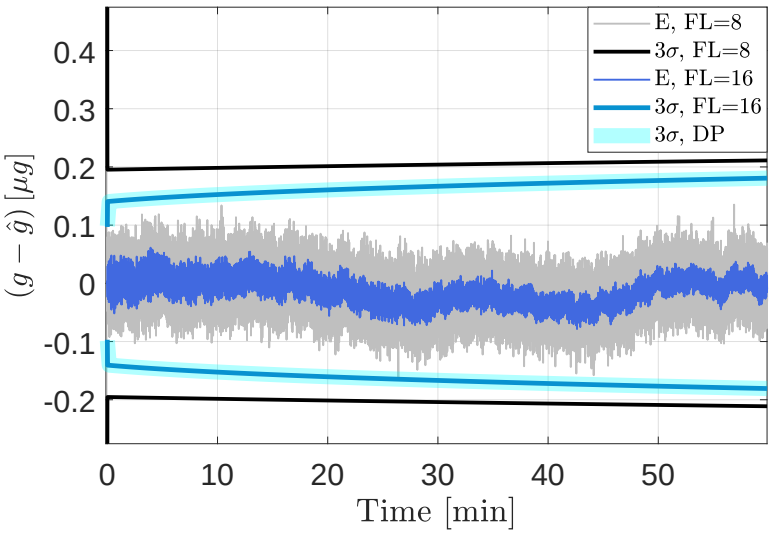
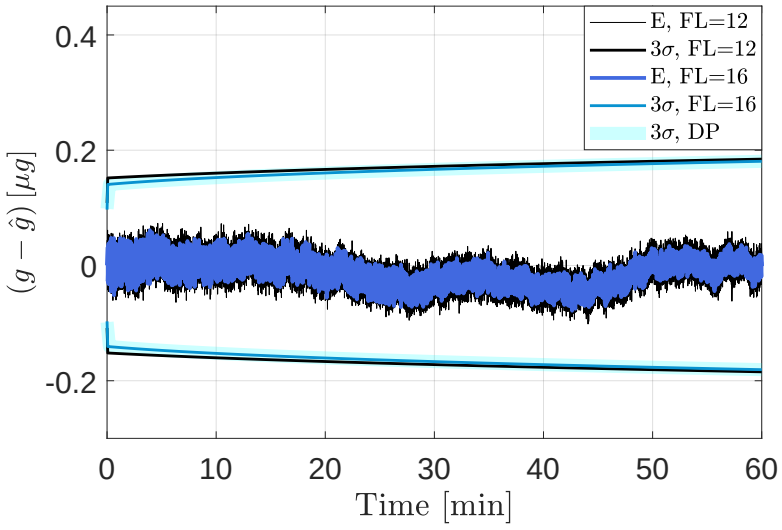


Figure 1. Acceleration estimation errors (E) and the corresponding 3σ bounds from the least squares-based moving average filter. The filter is run using 12 and 16-bit fractional length (FL) measurements. The 3σ bounds from double precision (DP) implementation of the least squares filter are also indicated.



(**a**)

Figure 2. *Cont.*



(**b**)

Figure 2. Acceleration estimation results from the quantized discrete-time Kalman filter (QDKF). The 3σ bounds from double precision (DP) simulations of the discrete-time Kalman filter without quantization errors are also shown for comparison. (**a**) Errors and corresponding 3σ bounds with quantized measurements of fractional lengths 8 and 16 bits. (**b**) Errors (E) and corresponding 3σ bounds with quantized measurements of fractional lengths of 12 and 16 bits.

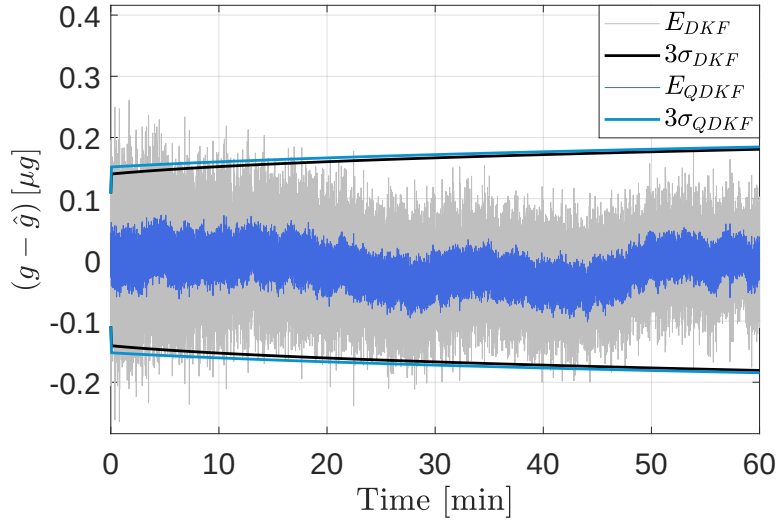


Figure 3. Acceleration estimation errors (E) and the corresponding 3σ bounds from the DKF and the QDKF. Measurements are quantized to fractional length of 12-bits.

4.1. Steady-State Performance

In practice, the Kalman filter is often run for long periods of time. As $k \to \infty$ and if given input remains within reasonable magnitude, the error covariance ($\mathbf{P}_k^-$) converges to a bounded steady-state value $\mathbf{P}$. For a large k, $\mathbf{P}_{k+1}^- = \mathbf{P}_k^- \triangleq \mathbf{P}$, and Equation (46) satisfies the discrete-time algebraic Riccati equation [24]:

$$\mathbf{P} = \widetilde{\mathbf{\Phi}}_k[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}\widetilde{\mathbf{\Phi}}_k^T + \widetilde{\mathbf{\Phi}}_k\mathbf{\Sigma}_{\boldsymbol{\chi},k}\widetilde{\mathbf{\Phi}}_k^T + \widetilde{\mathbf{\Gamma}}_k\mathbf{\Sigma}_{\hat{b},k}\widetilde{\mathbf{\Gamma}}_k^T + \begin{bmatrix}\tilde{\mathbf{Q}} & 0\\ 0 & \alpha\end{bmatrix} \tag{59}$$

with $\mathbf{K}_k$ given in Equation (47).

Figure 4 shows the standard deviation contours for steady-state acceleration estimation errors. These contours illustrate how the steady-state error bounds change as a function

of the fractional length in measurement quantization and the process noise associated with the acceleration channel. In the absence of quantization errors, the contour isolines maintain constant values for given levels of process noise. As the numerical precision of measurements increases, the contours asymptotically approach the steady-state performance achieved by floating-point implementation. The sensitivity analysis curves that graphically illustrate the impact of quantization noise on the state estimation accuracy are an important contribution of this work.

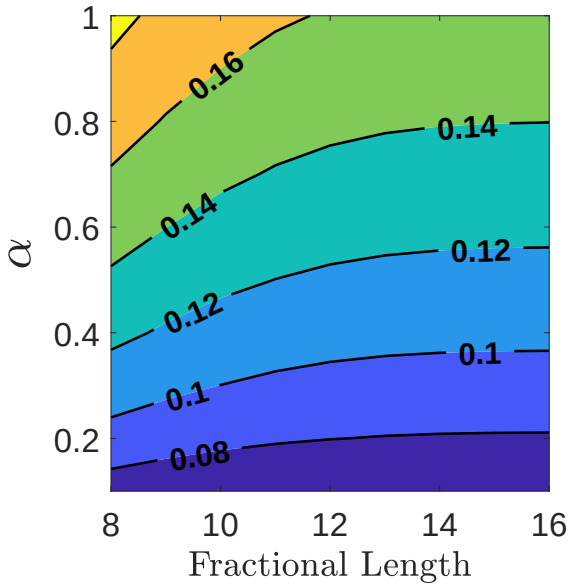


Figure 4. Steady-state 1σ contours for acceleration estimates. The contour lines are plotted as a function of measurement precision on the x-axis and the model uncertainty parameter for the acceleration channel, (α), on the y-axis.

The steady-state behavior reflects the filter performance over an extended period, depicting characteristics of the estimation errors once the transient effects have diminished and the error dynamics have stabilized. The steady-state analysis offers a quantitative measure of the best possible accuracy achievable by the filter and is based on sensor parameters, process, and measurement noise characteristics. Consequently, it serves as a valuable tool in sensor design and parameter tuning. In interferometric sensing, the design and modeling of mechanical elements, signal processing, and estimation filters can be tailored for accuracy and reliability using steady-state covariance analysis.

Furthermore, in the context of quantized Kalman filtering, the Mahalanobis distance is chosen as a metric to quantify the impact of quantization noise on the consistency of estimation errors. The Mahalanobis distance (d) between the quantized observation ($\mathbf{y}_k$) and its prediction ($\mathbf{H}_k\hat{\mathbf{X}}_k^-$)) reflects the variance between them, scaled by the inverse of the associated covariance matrix ($\mathbf{P}_{\mathbf{zz},k}^{-1}$). That is,

$$d = \sqrt{\mathbf{z}_k^T \mathbf{P}_{\mathbf{zz},k}^{-1} \mathbf{z}_k} \tag{60}$$

where, from Algorithm 1,

$$\mathbf{z}_k = \mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{X}}_k^- \tag{61}$$

$$\mathbf{P}_{\mathbf{zz},k} = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{H}_k \mathbf{\Sigma}_{\boldsymbol{\chi},k} \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{\Sigma}_{\mathbf{y},k} \tag{62}$$

Figure 5 illustrates the Mahalanobis distance of the estimates from the QDKF method for observations with varying fractional lengths over time. The Mahalanobis distance quantifies the dissimilarity between the estimated and true states, considering the covariance

of the estimation errors. A smaller Mahalanobis distance indicates better filter performance, as the estimated states are closer to the true states. To visualize the trend, a moving average of 10,000 Mahalanobis distance samples is computed. The results demonstrate that higher quantization noise in the observations leads to greater disparities between the observed and predicted measurements. While sequential filtering helps the filter learn the quantization noise statistics over time, it cannot fully match the lower bound achieved in double-precision simulations. Notably, this metric indicates that 16-bit measurements closely track the lower bound, suggesting an optimal word length for sensor data transmissions in this application.

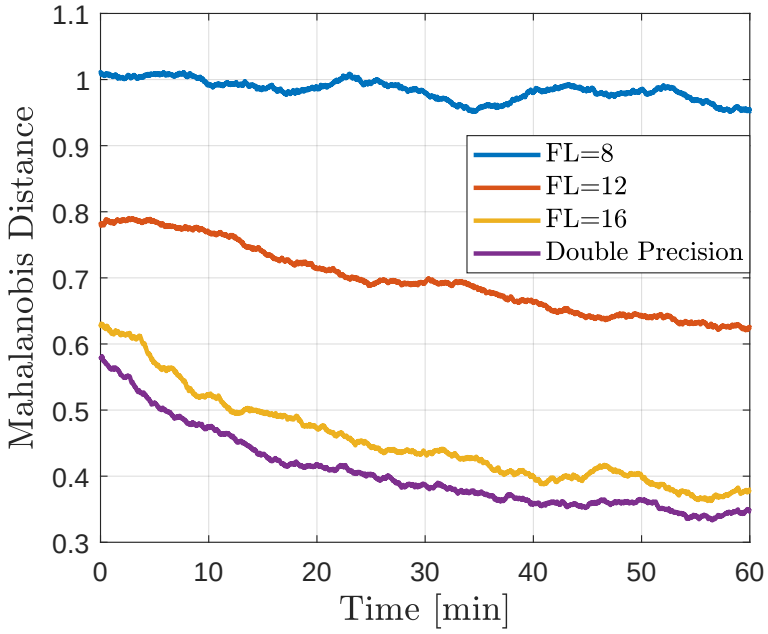


Figure 5. Mahalanobis distance moving average of 10,000 estimated samples.

4.2. Square Root Kalman Filter Simulations

For 12-bit measurements, the variances from the quantized DKF were observed to take negative values, which is theoretically nonviable. The SRKF, by construction, avoids loss of positive definiteness of the error covariance matrices and thereby offers resistance to numerical overflows and underflows. For covariance matrices that are appropriately scaled and quantized to avoid numerical degeneracy for this application, the performance of the QSRKF proposed in Algorithm 2 is comparable to that of the QDKF (Algorithm 1). Figure 6 compares the estimation errors and the corresponding 3σ bounds from QDKF and the QSRKF filters. Although this result indicates that the errors are not significantly reduced using the QSRKF version, the loss of positive definiteness in QDKF is a weakness of the standard implementation. Therefore, the square-root filters are preferable for onboard implementation even though they are computationally burdensome.

At the same time, it is evidently important to model quantization noise sources into the standard SRKF algorithm for reliable filter performance. Figure 7 compares the performance of the square-root filters with the quantization noise modeled into the filter structure (QSRKF in Algorithm 2) versus a standard SRKF implementation that does account for the round-off error statistics. In this comparison, the measurements are quantized to 12-bits in both the QSRKF and the SRKF runs.

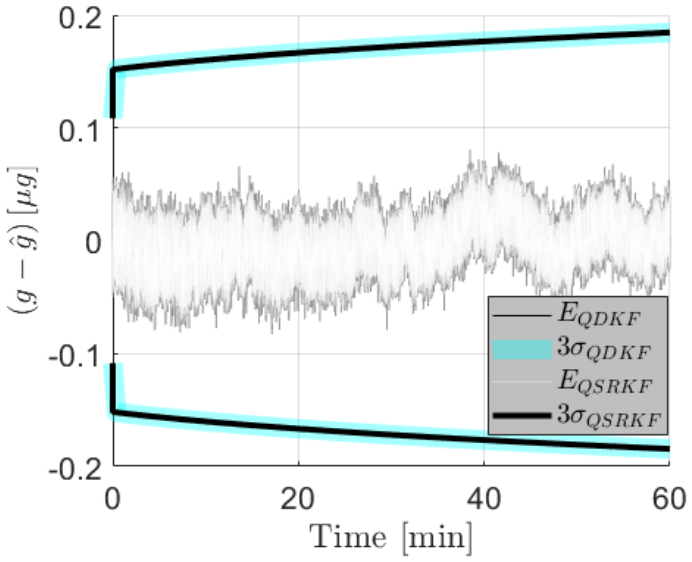


Figure 6. Acceleration state estimation errors (E) and the corresponding 3σ bounds from the QDKF and the QSRKF methods. Measurements are quantized to fractional length of 12-bits.

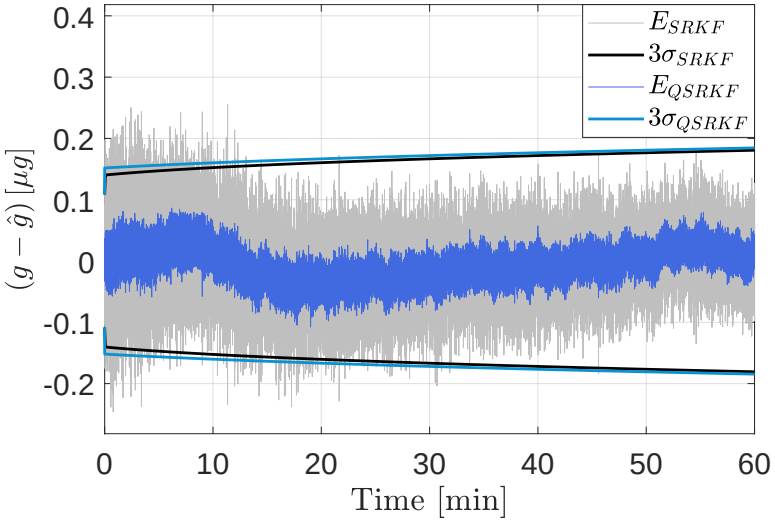


Figure 7. Acceleration estimation errors (E) and the corresponding 3σ bounds from the SRKF and the QSRKF. Measurements are quantized to fractional length of 12-bits.

Table 2 presents the time-averaged Mahalanobis distance for each filter (DKF, QDKF, SRKF, and QSRKF) as a function of measurement resolutions. The results show that the quantized filters (QDKF and QSRKF) proposed in this work outperform filter implementations that do not account for round-off errors (DKF and SRKF). The quantized filters maintain a lower average Mahalanobis distance, indicating better estimation accuracy and robustness to the effects of measurement quantization. Furthermore, the average Mahalanobis distance decreases as the fractional length increases for all filters. This trend suggests that higher measurement resolutions lead to improved filter performance, as more precise measurements provide better information for state estimation. The last column in the table (DP) represents the average Mahalanobis distance for the ideal filter implementations using double-precision measurements. These values serve as a benchmark for the best achievable performance without quantization effects. The quantized filters (QDKF and QSRKF) approach the performance of the ideal filters, demonstrating their effectiveness in mitigating the impact of measurement quantization.

Table 2. Average Mahalanobis distance for different filters and measurement fractional lengths (FLs). DP represents the ideal filter performance with double-precision measurements.

Filter	FL = 8	FL = 10	FL = 12	FL = 16	DP
DKF	36.3042	9.2255	2.4355	0.4855	0.4018
QDKF	0.9839	0.8647	0.6923	0.4584	
SRKF	36.3031	9.1854	2.4278	0.4827	0.4016
QSRKF	0.9839	0.8647	0.6923	0.4588	

Numerical simulations show that the Kalman filters appear to be under-confident because of the uncertainty in the evolution of the acceleration state. The least squares filter, however, does not require handling process noise associated with the acceleration state and is therefore much more confident about the estimation errors. Moreover, the process noise of the acceleration input is well studied to have low uncertainty [28]. Therefore, a least-squares based moving average filter is considered for hardware implementation.

5. Architecture for FPGA Implementation

In this section, an FPGA-based embedded architecture designed for estimating input acceleration force for a dual-oscillator system is described. The estimation algorithm consumes simulated measurements from two distinct oscillator models and delivers a covariance-weighted average of the estimated accelerations as illustrated in Figure 8.

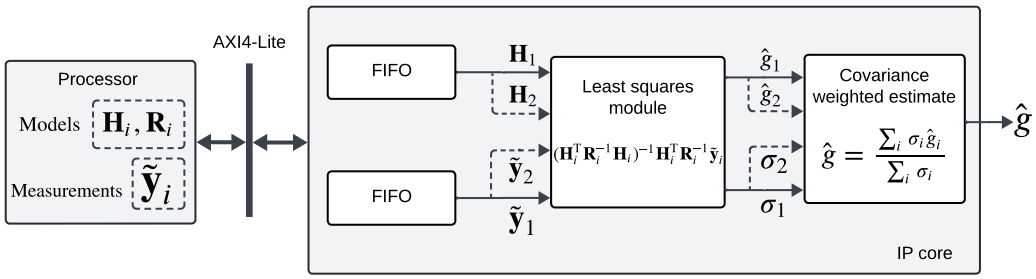


Figure 8. Block diagram for FPGA implementation of least squares-based covariance-weighted acceleration estimation method.

5.1. Implementation Overview

The system architecture comprises an FPGA accelerator designed to estimate the forcing accelerations from each of the oscillators within an FPGA sensor node and then to compute a covariance-weighted average of these estimates, as detailed in Section 3.4. The proposed architecture is a hardware-software co-design illustrated in Figure 8. Xilinx Zynq 7020 SoC is targeted for hardware evaluation of the proposed system. This SoC integrates a host processor, also known as the processing system (PS), featuring ARM Cortex-A9 MPCore. The PS performs the filter-specific operations that involve discrete-time state propagation and measurement model evaluation (Equations (27) and (40)). For a linear time-invariant system like the one under consideration, the state propagation from time-step t_k to t_{k+1} involves a constant state transition matrix and consequently a constant measurement model matrix (Equation (27)). Additionally, the processor manages the flow of simulated observations to be transmitted to the programmable logic (PL) via the memory-mapped register space indicated in Figure 9.

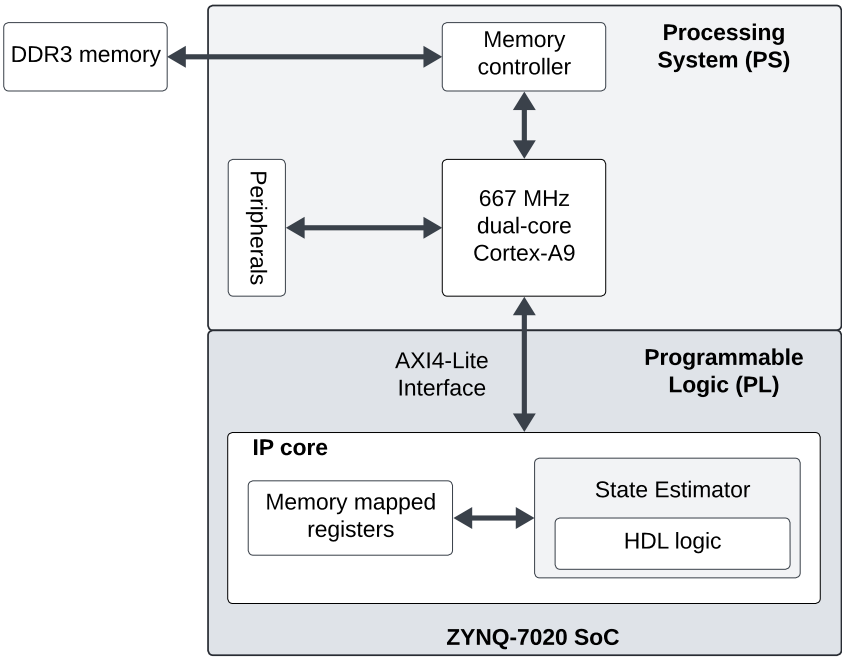


Figure 9. HW/SW codesign for Xilinx Zynq 7020 FPGA SoC-based acceleration state estimation: Filter-specific computations are implemented on the processing system (PS), and the filter-generic estimation algorithm is packaged as an intellectual property (IP) core and implemented on the programmable logic (PL). The 32-bit PS-PL on-chip communication bus is supported by the Advanced eXtensible Interface (AXI) protocol.

5.2. State Machine

The matrix operations in the intellectual property (IP) core are managed by a state machine shown in Figure 10. Initially, the IP core waits in an `IDLE` state, awaiting initialization by the processor. Upon receiving the initialization signal through a control register in the memory-mapped register space, the core transitions to the `INIT` state. Here, the processor initializes the internal memory (FIFO) of the IP core with values of the measurement model matrix augmented with the measurement noise covariance matrix. Once initialization is complete, signified by an `init_done` signal, the core progresses to the `LS_OPS` state, where least squares operations are executed. Upon completion of these operations, the core returns to the `IDLE` state, ready for the next batch of least squares operations, triggered by the arrival of another set of measurement model parameters relayed by the processor (for the second oscillator model in this case). This state machine governs the matrix operations within this pipelined least squares filter, with the hardware modules instantiated once and reused between different oscillator models.

In the subsequent stages of the implementation illustrated in Figure 8, the processor writes the simulated vector measurements into a data register, and the core performs the least squares estimation. Once estimates and corresponding variances from two such computations are available, the core calculates the covariance-weighted average of these estimates (Equation (38)). This result is then written by the core to a FIFO, ultimately to be read by the processor through another memory-mapped register.

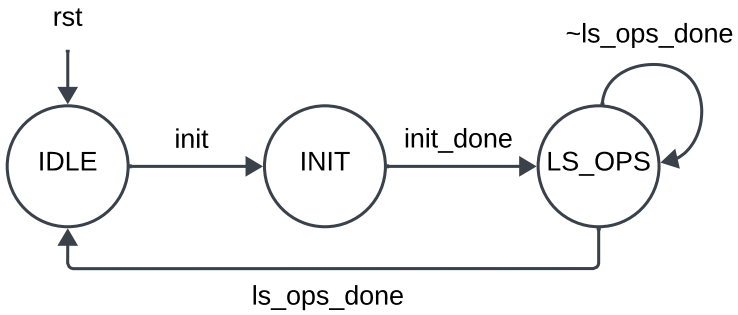


Figure 10. State machine for least squares operation.

5.3. Least Squares Module

This module implements a moving average filter that computes the normal equations using the direct expression for the least squares solution [23]. Operationally, three measurements are accumulated in a measurement vector $\tilde{\mathbf{y}}_i$ to compute the acceleration estimate. This implies that the dimension of matrix operations is three, with the estimates having a time delay equivalent to two measurement samples.

The estimation process involves a series of fixed-point matrix operations, including matrix transpose, inversion, matrix-matrix, and matrix-vector multiplications, to arrive at the least squares solution. Matrix-matrix multiplications are performed using systolic array architecture (SAA) [39], which is a pipelined two-dimensional network of multiply and accumulate (MAC) units, effective for low-latency matrix multiply operations. Matrix-vector multiplications also utilize MAC units to operate on the time-aligned input streams of matrix and measurement vector channels. The matrix inverse is computed using direct inversion expressions for the involved 3×3 matrices. The model and measurement data required for the computation are stored in the memory buffer of the IP core.

While the least squares solution in this example is tractable because the measurement model matrices are scaled to be well-conditioned, this approach can be computationally expensive. If $\mathbf{H}^T\mathbf{H}$ is ill-conditioned, the solution can be numerically unstable [40]. In practice, Cholesky (or) QR factorization [41], singular value decomposition, and, to a reasonable extent, LU decomposition [42] are efficient and accurate methods for solving normal equations.

The least squares module is instantiated only once, and the operations are pipelined to reuse the same module for estimating accelerations from two oscillators. This ensures effective resource management on the Zynq 7020 FPGA SoC.

5.4. Covariance Weighted Average Module

This module reads the buffered estimates and their corresponding variances to compute a weighted average of the acceleration estimate from two oscillator units, as described in Equation (38). Fixed-point division is performed using Xilinx's Divider Generator LogiCORE IP employing the radix-2 non-restoring integer division algorithm.

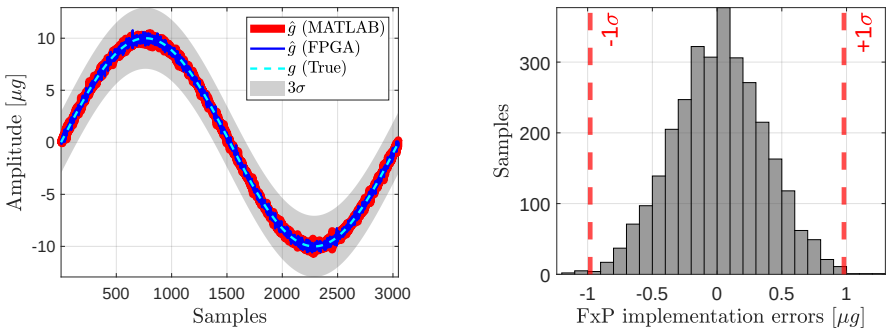
5.5. Implementation Results

The FPGA implementation of the estimation logic is validated using simulated inputs corresponding to the measurement models and the measurements themselves. To ensure accurate representation of values and avoid bit overflows and underflows, the inputs are scaled appropriately to maintain a consistent dynamic range. A MATLAB implementation of the same algorithm serves as a golden reference for comparison. The estimation results from the fixed-point FPGA implementation closely align with MATLAB's double-precision estimates and the ground truth, as depicted in Figure 11a. Although fewer, finite-precision numerical errors as high as 10% (up to 1 µg) are observed in the FPGA output. This is illustrated in Figure 11b. These round-off errors appear to propagate through the numerous matrix operations involved, resulting in a few outlier estimates. Adaptive

scaling, a higher number of fractional bits for data representation, and mixed-point (fixed and floating point) implementations are some techniques that could help reduce these errors. Additionally, solving normal equations through numerically robust methods such as Cholesky decomposition should further enhance estimation accuracy and potentially minimize the numerical errors.

5.6. Latency and Resource Utilization

The implementation results are shown in Table 3. This represents the implementation of state estimation logic deployed using SoC technology, where covariance-weighted acceleration estimation is implemented on an FPGA coupled to an ARM processor controlling the data flow. The hardware-software co-design required 188 digital signal processing (DSP) elements of the Zynq 7020 FPGA, running at 100 MHz. The low-latency co-design fits within 19% of the available FPGA look-up tables (LUTs), leaving the remaining resources for other potential application requirements such as sensor data processing. DSP usage can be reduced by further pipelining the operations or by strategically using LUTs for multiplications. The hardware-accelerated execution of pipelined acceleration estimation logic has an approximate latency of 3.62 µs for an estimation epoch from a dual-oscillator setup.



(**a**) (**b**)

Figure 11. Implementation results comparing fixed-point (FxP) FPGA output with floating-point MATLAB estimates. Acceleration estimation algorithm is the covariance-weighted average from dual-oscillator system (Section 3.4). (**a**) FxP FPGA output vs. double-precision MATLAB output. (**b**) Differences between FPGA and MATLAB estimates.

Table 3. Post-implementation FPGA resource utilization results.

Resource	Available	Utilization
LUTs	53,200	19%
LUTRAM	17,400	5%
Flip-Flops	106,400	14.78%
BRAM	140	0.71%
DSP	220	85.45%

6. Conclusions

In aerospace applications, system states are filtered on onboard embedded compute elements using measurements from a sensor network. The sensors and embedded flight computing systems are resource-constrained, limiting the precision of stored or transmitted data and consequently impacting the signal-to-noise ratio of the filter output. Accurate state estimation in finite-precision embedded implementations depends on the precision of the measurements and the word lengths of the state and input variables stored on the embedded computer.

This work presents an optomechanical sensor model for estimating forcing accelerations from simulated displacement measurements of a proof mass. The state estimation algorithms are reformulated to incorporate rounding errors into classical estimator frameworks. A least squares estimator, a discrete-time Kalman filter, and a square-root Kalman filter are developed for optimal state estimation with quantized measurement, state, and input variables. Numerical simulations demonstrate that the modified filter frameworks account for finite-precision effects, ensuring proper management of errors and uncertainties in the acceleration estimates. This approach maximizes the performance of filters implemented on fixed-point hardware architectures. Steady-state performance analysis shows that the best possible accuracy achievable by the filter is tightly coupled with the numerical precision of the internal variables. Metrics such as Mahalanobis distance give concrete insights into the word-length versus performance trade-offs.

Additionally, a dual-oscillator system for estimating acceleration states from independent measurements belonging to two oscillator models is proposed for hardware implementation. A covariance-weighted average of independent acceleration estimates is realized on an FPGA-SoC using a finite-precision implementation of the least squares method. The pipelined FPGA realization with simulated model and measurements reasonably tracks a double-precision MATLAB implementation of the same least squares-based estimation. In summary, this article addresses the realization of state estimation on embedded architectures, emphasizing the importance of managing finite word-length implementation errors to design and implement high-performance, resource-efficient estimation algorithms on memory-constrained computing systems.

It is worth noting that while this work thoroughly addresses quantization effects, the potential impacts of bit overflows are neglected. Scaling digital filter realizations to prevent overflow errors remains an avenue for future investigation.

Author Contributions: Conceptualization, R.R.B. and M.M.; methodology, R.R.B.; software, R.R.B.; validation, R.R.B. and M.M.; formal analysis, R.R.B. and M.M; investigation, R.R.B. and M.M.; resources, M.M. and F.G.; data curation, M.M. and F.G.; writing—original draft preparation, R.R.B.; writing—review and editing, R.R.B. and M.M; visualization, R.R.B.; supervision, M.M. and F.G.; project administration, M.M. and F.G.; funding acquisition, M.M. and F.G. All authors have read and agreed to the published version of the manuscript.

Funding: Support from the National Geospatial Intelligence Agency (NGA) through Grant No. HM0476-19-1-2015 is gratefully acknowledged. Dr. Scott A. True is thanked for serving as a technical monitor and research advocate for this work. Office of Naval Research Grant Number N00014-19-1-2435 is acknowledged for their support. The authors are thankful to Dr. Brian Holm-Hansen and Dr. David Gonzales for their encouragement and support. The Jet Propulsion Laboratory (JPL) subcontract 1659175 Texas Intelligent Space Systems (TISS) initiative and JPL's Strategic University Research Partnership (SURP) initiative are gratefully acknowledged.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset available on request from the authors.

Acknowledgments: The authors would like to thank Patrick Kelly and Adam Hines for their correspondence regarding their work on optomechanical inertial sensors. The authors also acknowledge the support and technical inputs from Anup Katake, Brian T. Young, Tim McElrath, Fred Y. Hadaegh, and Sarah U. Stevens from the NASA Jet Propulsion Laboratory.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DSP	Digital Signal Processing
DoF	Degree of Freedom
FPGA	Field Programmable Gate Array
SoC	System on Chip
COTS	Commercial Off-The-Shelf
QDKF	Quantized Discrete-time Kalman Filter
QSRKF	Quantized Square-Root Kalman Filter
PS	Processing System
PL	Programmable Logic
IP	Intellectual Property
AXI	Advanced eXtensible Interface
SAA	Systolic Array Architecture
LUT	Lookup Table
BRAM	Block Random Access Memory
HDL	Hardware Description Language

Appendix A. Least Squares Estimation with Quantized States and Measurements

In this section, the minimum variance estimation problem under the influence of quantization noise in states and measurements of a system is formulated. Following the state variable description developed by Mullis [26], Williamson and Kadiman [33], Liu and Skelton [10], and the least squares derivation by Crassidis and Junkins [35], the round-off errors are incorporated for minimum variance state estimation.

Appendix A.1. The Minimum Variance State Estimation Problem

Consider a discrete-time dynamical system:

$$\mathbf{x}_i = \mathbf{\Phi}(t_i, t_k)\mathbf{x}_k \tag{A1}$$

and the observational system given by:

$$\mathbf{z}_i = \tilde{\mathbf{H}}_i\mathbf{x}_i + \boldsymbol{\nu}_i \tag{A2}$$

where $\mathbb{E}\{\nu_i\} = \mathbf{0}\ \forall\, i$ and $\mathbb{E}\{\nu_i\nu_j^T\} = \mathbf{R}_{ij}$.

The above systems assume ideal i.e., infinite precision in states and measurements. However, in finite word length digital compute elements, the system states $\mathbf{x}_i$ and the measurements $\mathbf{z}_i$ will be quantized. A fixed-point finite word-length realization of the ideal systems in which quantization is implemented after accumulation of products is described by:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{\Phi}(t_i, t_k)\mathbb{Q}[\mathbf{x}_k] \\ \mathbf{z}_i &= \tilde{\mathbf{H}}_i\mathbb{Q}[\mathbf{x}_i] + \boldsymbol{\nu}_i \end{aligned} \tag{A3}$$

where $\mathbb{Q}[.]$ represents the quantization operator and $\mathbb{Q}[\mathbf{x}_i], \mathbb{Q}[\mathbf{z}_i]$ represent the quantized values of the system state and observation vectors, $\mathbf{x}_i, \mathbf{z}_i$, respectively.

Assuming the additive property of round-off errors, the quantization process affects the states and measurements as:

$$\begin{aligned} \mathbb{Q}[\mathbf{x}_i] &= \mathbf{x}_i + \boldsymbol{\epsilon}_{x,i} \\ \mathbb{Q}[\mathbf{z}_i] &= \mathbf{z}_i + \boldsymbol{\epsilon}_{z,i} \end{aligned} \tag{A4}$$

The round-off errors, $\boldsymbol{\epsilon}_{\mathbf{x},i}$ and $\boldsymbol{\epsilon}_{\mathbf{z},i}$, can be modeled as zero-mean, uncorrelated white noise sequences [34] with the error statistics described as:

$$\begin{aligned} &\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},i}\} = \mathbf{0} \quad \forall\, i \quad \text{and} \quad \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},i}\boldsymbol{\epsilon}_{\mathbf{x},j}^T\} = qI_{\mathbf{x}}; \quad q = \frac{2^{-2B_{\mathbf{x}}}}{12} \\ &\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{z},i}\} = \mathbf{0} \quad \forall\, i \quad \text{and} \quad \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{z},i}\boldsymbol{\epsilon}_{\mathbf{z},j}^T\} = qI_{\mathbf{z}}; \quad q = \frac{2^{-2B_{\mathbf{z}}}}{12} \\ &\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},i}\boldsymbol{\epsilon}_{\mathbf{z},i}^T\} = \mathbf{0} \end{aligned} \tag{A5}$$

The components $\mathbb{Q}[\mathbf{x}_i]$ and $\mathbb{Q}[\mathbf{z}_i]$ are assumed to be quantized to have $B_{\mathbf{x}}$-bit and $B_{\mathbf{z}}$-bit fractional representations, respectively. This assumption arises from the need to quantize the states to a desired finite-length ($B_{\mathbf{x}}$ bits) on the compute element and store them for subsequent calculations, while the measurements are quantized by an A/D converter. The model coefficients also have finite length but this implementation assumes rounding-off of the product of model coefficients and the state variables. For example, if $\tilde{\mathbf{H}}_i$ is quantized to B_c bits, the product $\tilde{\mathbf{H}}_i\mathbb{Q}[\mathbf{x}_i]$ results in a $B_{\mathbf{x}} + B_c$ bit fraction, and the result is quantized to $B_{\mathbf{x}}$ bits for subsequent operations. In essence, this approach does not directly optimize on the model coefficient errors but can only provide evaluation of the filter performance with respect to coefficient errors [10].

Substituting Equation (A4) into Equation (A3) and incorporating quantization in measurements to obtain the quantized state and observation systems as:

$$\begin{aligned} \mathbf{x}_i &= \boldsymbol{\Phi}(t_i, t_k)\mathbf{x}_k + \boldsymbol{\Phi}(t_i, t_k)\boldsymbol{\epsilon}_{\mathbf{x},k} \\ \mathbf{z}_i &= \tilde{\mathbf{H}}_i\mathbf{x}_i + \tilde{\mathbf{H}}_i\boldsymbol{\epsilon}_{\mathbf{x},i} + \boldsymbol{\nu}_i + \boldsymbol{\epsilon}_{\mathbf{z},i} \end{aligned} \tag{A6}$$

Here, processing a collection of measurements to estimate $\mathbf{x}_k$ through the use of state transition matrix amounts to:

$$\mathbf{z} = \mathbf{H}\mathbf{x}_k + \mathbf{H}\boldsymbol{\epsilon}_{\mathbf{x}} + \boldsymbol{\nu} + \boldsymbol{\epsilon}_{\mathbf{z}} \tag{A7}$$

where

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \tilde{\mathbf{H}}_1\boldsymbol{\Phi}(t_1, t_k) \\ \tilde{\mathbf{H}}_2\boldsymbol{\Phi}(t_2, t_k) \\ \vdots \end{bmatrix}, \quad \boldsymbol{\epsilon}_{\mathbf{x}} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{x},1} \\ \boldsymbol{\epsilon}_{\mathbf{x},2} \\ \vdots \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \vdots \end{bmatrix}, \quad \boldsymbol{\epsilon}_{\mathbf{z}} = \begin{bmatrix} \boldsymbol{\epsilon}_{\mathbf{z},1} \\ \boldsymbol{\epsilon}_{\mathbf{z},2} \\ \vdots \end{bmatrix}$$

A shorthand representation of Equation (A7) is:

$$\mathbf{z} = \mathbf{H}\mathbf{x}_k + \boldsymbol{\mu} \tag{A8}$$

where all the noise-related terms are combined into a new variable $\boldsymbol{\mu}$ such that

$$\boldsymbol{\mu} = \mathbf{H}\boldsymbol{\epsilon}_{\mathbf{x}} + \boldsymbol{\nu} + \boldsymbol{\epsilon}_{\mathbf{z}} \tag{A9}$$

Before deriving the estimator, let's define the statistics of measurement and quantization noises in Equation (A9). For the collection of measurements, the first and the second central moments are described as:

$$\mathbb{E}\{\boldsymbol{\nu}\} = [\mathbf{0} \quad \mathbf{0} \quad \dots]^T \quad \text{and} \quad \mathbb{E}\{\boldsymbol{\nu}\boldsymbol{\nu}^T\} = \mathbf{P}_{\nu\nu} \tag{A10}$$

The central moments of the quantization noise in states and the measurements are described in Equation (A5) and noted here as:

$$\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x}}\} = [\mathbf{0} \quad \mathbf{0} \quad \dots]^T \quad \text{and} \quad \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x}}\boldsymbol{\epsilon}_{\mathbf{x}}^T\} = \boldsymbol{\Sigma}_{\mathbf{x}} \tag{A11}$$

$$\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{z}}\} = [\mathbf{0} \quad \mathbf{0} \quad \dots]^T \quad \text{and} \quad \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{z}}\boldsymbol{\epsilon}_{\mathbf{z}}^T\} = \boldsymbol{\Sigma}_{\mathbf{z}} \tag{A12}$$

Finally, the mean and covariance of the combined noise model $\boldsymbol{\mu}$ is:

$$\begin{aligned}\mathbb{E}\{\boldsymbol{\mu}\} &= \mathbb{E}\{\mathbf{H}\boldsymbol{\epsilon}_{\mathbf{x}} + \boldsymbol{\nu} + \boldsymbol{\epsilon}_{\mathbf{z}}\} \\ &= \mathbf{H}\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x}}\} + \mathbb{E}\{\boldsymbol{\nu}\} + \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{z}}\} \\ &= \mathbf{H}.\mathbf{0} + \mathbf{0} + \mathbf{0} = \mathbf{0}\end{aligned} \tag{A13}$$

$$\begin{aligned}\mathbf{P}_{\mu\mu} &= \mathbb{E}\{\boldsymbol{\mu}\boldsymbol{\mu}^T\} \\ &= \mathbb{E}\{\left[\mathbf{H}\boldsymbol{\epsilon}_{\mathbf{x}} + \boldsymbol{\nu} + \boldsymbol{\epsilon}_{\mathbf{z}}\right]\left[\mathbf{H}\boldsymbol{\epsilon}_{\mathbf{x}} + \boldsymbol{\nu} + \boldsymbol{\epsilon}_{\mathbf{z}}\right]^T\}\end{aligned} \tag{A14}$$

wherein identifying that the errors are mutually uncorrelated and the definitions of individual covariances follows that

$$\begin{aligned}\mathbf{P}_{\mu\mu} &= \mathbf{H}\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x}}\boldsymbol{\epsilon}_{\mathbf{x}}^T\}\mathbf{H}^T + \mathbb{E}\{\boldsymbol{\nu}\boldsymbol{\nu}^T\} + \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{z}}\boldsymbol{\epsilon}_{\mathbf{z}}^T\} \\ &= \mathbf{H}\boldsymbol{\Sigma}_{\mathbf{x}}\mathbf{H}^T + \mathbf{P}_{\nu\nu} + \boldsymbol{\Sigma}_{\mathbf{z}}\end{aligned} \tag{A15}$$

Appendix A.2. Linear, Unbiased, Minimum Variance Estimation

The objective of the estimator is to seek a linear, unbiased, minimum variance estimate, $\hat{\mathbf{x}}_k$ of the state $\mathbf{x}_k$.

Linear. To begin with, the desired estimate, $\hat{\mathbf{x}}_k$, is a linear combination of measurements. That is

$$\hat{\mathbf{x}}_k = \mathbf{M}\mathbf{z} + \mathbf{n} \tag{A16}$$

where, for n number of states and m number of measurements, an optimal choice of $\mathbf{M} \in \mathbb{R}^{n\times m}$ and $\mathbf{n} \in \mathbb{R}^{n\times 1}$ is to be determined.

Additionally, for a perfect set of measurements i.e., $\boldsymbol{\nu}_i = \mathbf{0}\ \forall\ i$, and a perfect measurement model $\mathbf{z} = \mathbf{H}\mathbf{x}(k)$, the measurement system in Equation (A7) should result in the true state $\mathbf{x}(k)$ such that

$$\begin{aligned}\hat{\mathbf{x}}_k = \mathbf{x}_k &= \mathbf{M}\mathbf{z} + \mathbf{n} \\ \mathbf{x}_k &= \mathbf{M}\mathbf{H}\mathbf{x}_k + \mathbf{n}\end{aligned} \tag{A17}$$

Unbiased. Next, for an unbiased estimate, the expected value of the estimated state should be the true state. This, combined with the linear model assumption in Equation (A16), gives:

$$\mathbb{E}\{\hat{\mathbf{x}}_k\} = \mathbf{x}_k \tag{A18}$$

$$\mathbb{E}\{\mathbf{M}\mathbf{z} + \mathbf{n}\} = \mathbf{x}_k \tag{A19}$$

Then, from the assumed measurement model, it follows that

$$\mathbb{E}\{\mathbf{M}[\mathbf{H}\mathbf{x}_k + \boldsymbol{\mu}] + \mathbf{n}\} = \mathbb{E}\{\mathbf{M}\mathbf{H}\mathbf{x}_k + \mathbf{M}\boldsymbol{\mu} + \mathbf{n}\} = \mathbf{x}_k \tag{A20}$$

Since, the matrices $\mathbf{M}$, $\mathbf{H}$ and the vector $\mathbf{n}$ are deterministic and the noise is a zero-mean process (Equation (A13)), it follows that

$$\begin{aligned}\mathbf{M}\mathbf{H}\mathbb{E}\{\mathbf{x}_k\} + \mathbf{M}\mathbb{E}\{\boldsymbol{\mu}\} + \mathbf{n} &= \mathbf{x}_k \\ \mathbf{M}\mathbf{H}\mathbf{x}_k + \mathbf{n} &= \mathbf{x}_k\end{aligned} \tag{A21}$$

where $\mathbf{M}$ and $\mathbf{n}$ satisfy the constraints:

$$\mathbf{M}\mathbf{H} = \mathbf{I}_n \quad \text{and} \quad \mathbf{n} = 0 \tag{A22}$$

Minimum Variance. As a final condition, an optimal minimum variance estimator is the one that has the smallest variance of all possible estimators. The objective is to minimize the state covariance to find an optimal choice of $\mathbf{M}$.

If $\mathbf{e}_{\mathbf{x}k}$ defines the state estimation error such as $\mathbf{e}_{\mathbf{x}k} = \mathbf{x}_k - \hat{\mathbf{x}}_k$, the error mean and covariances are identified as:

$$\mathbb{E}\{\mathbf{e}_{\mathbf{x},k}\} = \mathbb{E}\{\mathbf{x}_k - \hat{\mathbf{x}}_k\} = \mathbb{E}\{\mathbf{x}_k\} - \mathbb{E}\{\hat{\mathbf{x}}_k\} = \mathbf{0} \tag{A23}$$

$$\mathbf{P}_{\mathbf{xx},k} = \mathbb{E}\{\left[\mathbf{x}_k - \hat{\mathbf{x}}_k\right]\left[\mathbf{x}_k - \hat{\mathbf{x}}_k\right]^T\} \tag{A24}$$

Using the quantized observational system (Equation (A7)) and the linear estimation model (Equation (A16)) in the above expression for covariance gives:

$$\begin{aligned}\mathbf{P}_{\mathbf{xx},k} &= \mathbb{E}\{\left[\mathbf{x}_k - \mathbf{M}\mathbf{z}\right]\left[\mathbf{x}_k - \mathbf{M}\mathbf{z}\right]^T\} \\ &= \mathbb{E}\{\left[\mathbf{x}_k - \mathbf{M}(\mathbf{H}\mathbf{x}_k + \boldsymbol{\mu})\right]\left[\mathbf{x}_k - \mathbf{M}(\mathbf{H}\mathbf{x}_k + \boldsymbol{\mu})\right]^T\} \\ &= \mathbf{M}\mathbf{P}_{\mu\mu}\mathbf{M}^T\end{aligned} \tag{A25}$$

where the constraint $\mathbf{M}\mathbf{H} = \mathbf{I}_n$ is used to eliminate $\mathbf{x}_k$'s in the second step while the fact that $\mathbf{M}$ is deterministic is used in the second last step.

Constrained Optimization

The goal is to minimize the covariance matrix $\mathbf{P}_{\mathbf{xx},k}$ to determine $\mathbf{M}$ while respecting the constraint on it. The covariance to minimize becomes:

$$\mathbf{P}_{\mathbf{xx},k} = \mathbf{M}\mathbf{P}_{\mu\mu}\mathbf{M}^T + \boldsymbol{\Lambda}^T[\mathbf{I}_n - \mathbf{M}\mathbf{H}]^T + [\mathbf{I}_n - \mathbf{M}\mathbf{H}]\boldsymbol{\Lambda} \tag{A26}$$

wherein the constraint on $\mathbf{M}$ is accounted using a matrix of Lagrange multipliers, $\boldsymbol{\Lambda}$, and the fact that $\mathbf{P}_{\mathbf{xx},k}$ is symmetric is respected by adding the transpose of the constraint term.

Setting the first variation of $\mathbf{P}_{\mathbf{xx},k}$ in the above equation to zero i.e., $\delta\mathbf{P}_{\mathbf{xx},k} = 0$, gives two simultaneous conditions:

$$\mathbf{M}\mathbf{P}_{\mu\mu} - \boldsymbol{\Lambda}^T\mathbf{H}^T = 0 \quad \text{and} \tag{A27}$$

$$\mathbf{I}_n - \mathbf{M}\mathbf{H} = 0 \tag{A28}$$

Here $\mathbf{P}_{\mu\mu}$ is a positive definite matrix, therefore from the above equation $\mathbf{M}$ is determined as:

$$\mathbf{M} = \boldsymbol{\Lambda}^T\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1} \tag{A29}$$

substituting $\mathbf{M}$ in Equation (A28) gives an expression for the Lagrange multiplier matrix as:

$$\boldsymbol{\Lambda}^T = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}]^{-1} \tag{A30}$$

Using the Equation (A30) in Equation (A29), the optimal choice of $\mathbf{M}$ that minimizes the covariance $\mathbf{P}_{\mathbf{xx}}(k)$, is obtained as:

$$\mathbf{M} = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1} \tag{A31}$$

Finally, the unbiased estimate $\hat{\mathbf{x}}(k)$ that is linear in measurements $\mathbf{z}$ is given as:

$$\begin{aligned}\hat{\mathbf{x}}_k &= \mathbf{M}\mathbf{z} \\ \hat{\mathbf{x}}_k &= [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{z}\end{aligned} \tag{A32}$$

Also, the covariance matrix can be derived by substituting $\mathbf{M}$ in $\mathbf{P}_{\mathbf{xx}}(k) = \mathbf{M}\mathbf{P}_{\mu\mu}\mathbf{M}^T$ as:

$$\mathbf{P}_{\mathbf{xx},k} = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}]^{-1} \tag{A33}$$

This derivation is similar to the classical minimum variance estimator with measurement errors. However, from re-deriving the estimator in the presence of round-off errors in states and measurements, the unbiased estimates and estimation covariance are observed to be interestingly impacted. Firstly, while the least squares filter structure appears to remain unaffected by round-off errors, it necessitates careful attention to the elements involved in the structure. Notably, the covariance matrix is now influenced by the state quantization noise that is propagated through the system dynamics. The measurement quantization errors further expand the covariance matrix. The resulting least squares filter serves as an optimal estimator of state nodes quantized after each accumulation epoch, derived from quantized measurements. The filter expressions are summarized in Table A1.

Table A1. Quantized minimum variance estimator.

State estimate	$\hat{\mathbf{x}}_k = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}]^{-1}\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{z}$
State error covariance	$\mathbf{P}_{\mathbf{xx},k} = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H}]^{-1}$
Measurement error covariance	$\mathbf{P}_{\mu\mu} = \mathbf{H}\mathbf{\Sigma}_\mathbf{x}\mathbf{H}^T + \mathbf{P}_{\nu\nu} + \mathbf{\Sigma}_\mathbf{z}$

Appendix A.3. Estimation with a Priori Information

In the least squares formulation of Equation (A32), the *a priori* knowledge of the model dynamics can be rigorously incorporated to obtain an updated state estimate based on the recently seen information. The same formulation as that of the least squares filter can be extended to incorporate prior information about the state [35]. Thereby, a maximum *a posteriori* estimate under the presence of *a priori* state information $\hat{\mathbf{x}}_a$ and an *a priori* error covariance matrix $\mathbf{Q}_a$ can be written as:

$$\hat{\mathbf{x}}(k) = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H} + \mathbf{Q}_a^{-1}]^{-1}[\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{z} + \mathbf{Q}_a^{-1}\hat{\mathbf{x}}_{a,k}] \tag{A34}$$

$$\mathbf{P}_{\mathbf{xx},k} = [\mathbf{H}^T\mathbf{P}_{\mu\mu}^{-1}\mathbf{H} + \mathbf{Q}_a^{-1}]^{-1} \tag{A35}$$

Appendix B. Quantized Discrete-Time Kalman Filter (QDKF)

Practical implementation of the Kalman filter often encounters numerous challenges. One frequent issue is due to filter divergence when actual estimation errors statistically deviate from computed estimation errors [43]. This discrepancy can lead to estimation errors exceeding the confidence intervals defined by the computed error covariance, which may approach infinity. Consequently, the filter fails to reach a steady state, especially as the measurement interval tends to infinity. Such anomalies may stem from errors in the system model upon which the filter relies or inaccuracies in modeling the error statistics [44].

This work introduces a theoretical framework for modeling errors in Kalman filter implementation on finite-precision hardware. The derivation involves incorporating quantization noises into the filter design. It will be demonstrated that integrating quantization errors into the model renders the resulting filter sensitive to the embedded architecture on which it is implemented. Moreover, embedding such errors enhances the filter's ability to recover from otherwise unmodeled errors.

Appendix B.1. Kalman Filter Derivation

Following the assumptions on quantization noise properties that are put forth in the least squares formulation (Appendix A.1), this derivation considers a linear, discrete-time dynamical system defined by the difference equation as:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k\mathbf{x}_k + \mathbf{\Gamma}_k\mathbf{u}_k + \gamma_k\mathbf{w}_k \tag{A36}$$

where, at time k, $\mathbf{x}_k$ is the system state, $\mathbf{u}_k$ is the control input, $\mathbf{w}_k$ is additive process noise, $\mathbf{\Phi}_k$ is the state transition matrix, $\mathbf{\Gamma}_k$ is the input transition matrix, and γ_k is a deterministic matrix that maps process noise into the state dynamics.

Further, the measurements are linearly related to the states and are available in discrete-time form as:

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{A37}$$

where $\mathbf{y}_k$ is the measurement obtained at time k, $\mathbf{H}_k$ is the observation matrix, and $\mathbf{v}_k$ represents the additive noise that corrupts the measurements. Note that $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be uncorrelated zero-mean Gaussian white-noise processes with variances defined as:

$$\begin{aligned} \mathbb{E}\{\mathbf{w}_k \mathbf{w}_l^T\} &= \mathbf{Q}_k \delta_{kl} \\ \mathbb{E}\{\mathbf{v}_k \mathbf{v}_l^T\} &= \mathbf{R}_k \delta_{kl} \end{aligned} \qquad \delta_{kl} = \begin{cases} 1, & \text{if } k = l. \\ 0, & \text{if } k \neq l. \end{cases} \tag{A38}$$

Furthermore, it is supposed that the initial system state has a known mean and covariance, $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$ respectively defined as:

$$\hat{\mathbf{x}}_0 = \mathbb{E}[\mathbf{x}_0] \quad \text{and} \quad \mathbf{P}_0 = \mathbb{E}[(\hat{\mathbf{x}}_0 - \mathbf{x}_0)(\hat{\mathbf{x}}_0 - \mathbf{x}_0)^T] \tag{A39}$$

Appendix B.1.1. Quantization Assumptions

In the finite-precision implementation of the Kalman filter, the round-off errors are to be modeled in the filter derivation. Assuming that quantization ($\mathbb{Q}[.]$) is implemented at state nodes after double-length accumulation, and that measurements from A/D conversion as well as input resulting from D/A conversion are independently quantized, the additional quantization errors effect the states, measurements and inputs as follows:

$$\mathbb{Q}[\mathbf{x}_k] = \mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k} \tag{A40}$$

$$\mathbb{Q}[\mathbf{y}_k] = \mathbf{y}_k + \boldsymbol{\epsilon}_{\mathbf{y},k} \tag{A41}$$

$$\mathbb{Q}[\mathbf{u}_k] = \mathbf{u}_k + \boldsymbol{\epsilon}_{\mathbf{u},k} \tag{A42}$$

Following the description in Equation (A5), the round-off errors $\boldsymbol{\epsilon}_{\mathbf{x},k}$, $\boldsymbol{\epsilon}_{\mathbf{y},k}$, and $\boldsymbol{\epsilon}_{\mathbf{u},k}$, can be modeled as zero-mean, uncorrelated white noise process with the error statistics defined as:

$$\begin{aligned} \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},k}\} &= \mathbf{0} \quad \forall i \quad \text{and} \quad \mathbf{\Sigma}_{\mathbf{x},k} = \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},k} \boldsymbol{\epsilon}_{\mathbf{x},l}^T\} = q I_{\mathbf{x}}; \quad q = \frac{2^{-2B_{\mathbf{x}}}}{12} \\ \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{y},k}\} &= \mathbf{0} \quad \forall i \quad \text{and} \quad \mathbf{\Sigma}_{\mathbf{y},k} = \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{y},k} \boldsymbol{\epsilon}_{\mathbf{y},l}^T\} = q I_{\mathbf{y}}; \quad q = \frac{2^{-2B_{\mathbf{y}}}}{12} \\ \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{u},k}\} &= \mathbf{0} \quad \forall i \quad \text{and} \quad \mathbf{\Sigma}_{\mathbf{u},k} = \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{u},k} \boldsymbol{\epsilon}_{\mathbf{u},l}^T\} = q I_{\mathbf{u}}; \quad q = \frac{2^{-2B_{\mathbf{u}}}}{12} \end{aligned} \tag{A43}$$

These additional errors for finite-precision implementation can be modeled into the filter description in Equations (A36) and (A37) as:

$$\mathbf{x}_{k+1} = \mathbf{\Phi}_k (\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{\Gamma}_k (\mathbf{u}_k + \boldsymbol{\epsilon}_{\mathbf{u},k}) + \gamma_k \mathbf{w}_k \tag{A44}$$

$$\mathbf{y}_k = \mathbf{H}_k (\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k} \tag{A45}$$

Appendix B.1.2. Derivation

Given the above assumptions, the objective is to determine an optimal estimate of the state at $(k+1)^{\text{th}}$ instance i.e., $\hat{\mathbf{x}}_{k+1}$ based upon a set of $k+1$ sets of measurements and the current state estimate at k. The Kalman filter comprises of propagation and update stages for states and covariances. The propagation stage predicts *a priori* mean and error

covariance of the state while the update stage uses new measurements to update the prediction to yield *a posteriori* mean and error covariance.

The propagation of the state is attained by taking an expected value of the difference equation in Equation (A44). The mean and covariances are obtained as follows:

$$\begin{aligned}\hat{\mathbf{x}}_{k+1} &= \mathbb{E}[\mathbf{\Phi}_k\mathbf{x}_k + \mathbf{\Phi}_k\boldsymbol{\epsilon}_{\mathbf{x},k} + \mathbf{\Gamma}_k\mathbf{u}_k + \mathbf{\Gamma}_k\boldsymbol{\epsilon}_{\mathbf{u},k} + \gamma_k\mathbf{w}_k] \\ &= \mathbf{\Phi}_k\mathbb{E}[\mathbf{x}_k] + \mathbf{\Gamma}_k\mathbb{E}[\mathbf{u}_k] \\ \hat{\mathbf{x}}_{k+1} &= \mathbf{\Phi}_k\hat{\mathbf{x}}_k + \mathbf{\Gamma}_k\mathbf{u}_k \end{aligned} \tag{A46}$$

In arriving at this result, deterministic nature of the transition matrices and the input vector as well as zero-mean properties of the process and quantization noises is utilized. The covariance propagation stems from the state estimation errors at discrete instances $k+1$ and k determined as:

$$\begin{aligned}\mathbf{e}_{\mathbf{x},k+1} &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1} \\ &= [\mathbf{\Phi}_k(\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{\Gamma}_k(\mathbf{u}_k + \boldsymbol{\epsilon}_{\mathbf{u},k}) + \gamma_k\mathbf{w}_k] - [\mathbf{\Phi}_k\hat{\mathbf{x}}_k + \mathbf{\Gamma}_k\mathbf{u}_k] \\ &= \mathbf{\Phi}_k(\mathbf{e}_{\mathbf{x},k} + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{\Gamma}_k\boldsymbol{\epsilon}_{\mathbf{u},k} + \gamma_k\mathbf{w}_k \end{aligned} \tag{A47}$$

where $\mathbf{e}_{\mathbf{x},k} = \mathbf{x}_k - \hat{\mathbf{x}}_k$.

Now the estimation error covariance at $(k+1)^{\text{th}}$ instance, $\mathbf{P}_{k+1}$, is determined using the above error propagation equation as:

$$\begin{aligned}\mathbf{P}_{k+1} &= \mathbb{E}\{\mathbf{e}_{\mathbf{x},k+1}\mathbf{e}_{\mathbf{x},k+1}^T\} \\ &= \mathbb{E}\{[\mathbf{\Phi}_k(\mathbf{e}_{\mathbf{x},k} + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{\Gamma}_k\boldsymbol{\epsilon}_{\mathbf{u},k} + \gamma_k\mathbf{w}_k] + [\mathbf{\Phi}_k(\mathbf{e}_{\mathbf{x},k} + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{\Gamma}_k\boldsymbol{\epsilon}_{\mathbf{u},k} + \gamma_k\mathbf{w}_k]^T\} \\ &= \mathbf{\Phi}_k\mathbb{E}\{(\mathbf{e}_{\mathbf{x},k} + \boldsymbol{\epsilon}_{\mathbf{x},k})(\mathbf{e}_{\mathbf{x},k} + \boldsymbol{\epsilon}_{\mathbf{x},k})^T\}\mathbf{\Phi}_k^T + \mathbf{\Gamma}_k\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{u},k}\boldsymbol{\epsilon}_{\mathbf{u},k}^T\}\mathbf{\Gamma}_k^T + \gamma_k\mathbb{E}\{\mathbf{w}_k\mathbf{w}_k^T\}\gamma_k^T \\ \mathbf{P}_{k+1} &= \mathbf{\Phi}_k(\mathbf{P}_k + \mathbf{\Sigma}_{\mathbf{x},k})\mathbf{\Phi}_k^T + \mathbf{\Gamma}_k\mathbf{\Sigma}_{\mathbf{u},k}\mathbf{\Gamma}_k^T + \gamma_k\mathbf{Q}_k\gamma_k^T \end{aligned} \tag{A48}$$

wherein the above expression is obtained from the information that $\mathbf{\Phi}_k, \mathbf{\Gamma}_k$ and γ_k are deterministic., the noise terms are uncorrelated as defined in the Equations (A38) and (A43).

Completing the propagation step gives prior mean and error covariance of the state which from now on be called as $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$, respectively. After the propagation step, the new measurement information updates the estimated state and the confidence in that estimate. In the step it is desired to update the prior estimate of the state and error covariance $\{\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-\}$ to produce *a posteriori* mean and covariance, $\{\hat{\mathbf{x}}_k^+, \mathbf{P}_k^+\}$. A linear update equation for the new estimate combines the prior estimate with measurement data and can be expressed as:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{A49}$$

$$\hat{\mathbf{x}}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{y}_k \tag{A50}$$

where $\mathbf{K}_k$ is the Kalman gain which multiplies innovation and adds it to the previous best estimate of the state. The difference between the actual measurement and its prediction i.e., $(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$, is known as the innovation, $\mathbf{z}_k$. The expression for the covariance of the innovation, $\mathbf{P}_{\mathbf{zz},k}$, will be derived shortly.

From the update equation in Equation (A45) and the measurement model described in Equation (A49), the posterior estimation error can be computed as:

$$\begin{aligned}
\mathbf{e}_{\mathbf{x},k}^{+} &= \mathbf{x}_k - \hat{\mathbf{x}}_k^{+} \\
&= \mathbf{x}_k - ([\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\hat{\mathbf{x}}_k^{-} + \mathbf{K}_k\mathbf{y}_k) \\
&= \mathbf{x}_k - \left([\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\hat{\mathbf{x}}_k^{-} + \mathbf{K}_k[\mathbf{H}_k(\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}]\right) \\
&= [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k](\mathbf{x}_k - \hat{\mathbf{x}}_k^{-}) - \mathbf{K}_k[\mathbf{H}_k\boldsymbol{\epsilon}_{\mathbf{x},k} + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}] \\
\mathbf{e}_{\mathbf{x},k}^{+} &= [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{e}_{\mathbf{x},k}^{-} - \mathbf{K}_k[\mathbf{H}_k\boldsymbol{\epsilon}_{\mathbf{x},k} + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}]
\end{aligned} \tag{A51}$$

Now, the posterior estimation error covariance from the posterior state estimation error is obtained as:

$$\begin{aligned}
\mathbf{P}_k^{+} &= \mathbb{E}\{\mathbf{e}_{\mathbf{x},k}^{+}\mathbf{e}_{\mathbf{x},k}^{+\,T}\} \\
&= \mathbb{E}\Big\{\left([\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{e}_{\mathbf{x},k}^{-} - \mathbf{K}_k[\mathbf{H}_k\boldsymbol{\epsilon}_{\mathbf{x},k} + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}]\right) \\
&\qquad \left([\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{e}_{\mathbf{x},k}^{-} - \mathbf{K}_k[\mathbf{H}_k\boldsymbol{\epsilon}_{\mathbf{x},k} + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}]\right)^{T}\Big\} \\
&= [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbb{E}\{\mathbf{e}_{\mathbf{x},k}^{-}\mathbf{e}_{\mathbf{x},k}^{-\,T}\}[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \\
&\qquad \mathbf{K}_k\Big[\mathbf{H}_k\mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{x},k}\boldsymbol{\epsilon}_{\mathbf{x},k}^T\}\mathbf{H}_k^T + \mathbb{E}\{\mathbf{v}_k\mathbf{v}_k^T\} + \mathbb{E}\{\boldsymbol{\epsilon}_{\mathbf{y},k}\boldsymbol{\epsilon}_{\mathbf{y},k}^T\}\Big]\mathbf{K}_k^T \\
\mathbf{P}_k^{+} &= [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^{-}[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \mathbf{K}_k\Big[\mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{x},k}\mathbf{H}_k^T + \mathbf{R}_k + \boldsymbol{\Sigma}_{\mathbf{y},k}\Big]\mathbf{K}_k^T
\end{aligned} \tag{A52}$$

Equation (A52) is the posterior error covariance update that is derived assuming the Kalman gain $\mathbf{K}_k$ to be deterministic.

In order to determine $\mathbf{K}_k$, the mean squared error of state estimation error is minimized by minimizing the trace of $\mathbf{P}_k^{+}$. In other words, the trace of $\mathbf{P}_k^{+}$ is differentiated with respect to $\mathbf{K}_k$ and the result is set to zero in search of the minimizing conditions. That is

$$\frac{\partial \mathrm{Tr}(\mathbf{P}_k^{+})}{\partial \mathbf{K}_k} = \mathbf{0} = -2[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^{-}\mathbf{H}_k^T + 2\mathbf{K}_k[\mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{x},k}\mathbf{H}_k^T + \mathbf{R}_k + \boldsymbol{\Sigma}_{\mathbf{y},k}] \tag{A53}$$

where the following trace identities are used

$$\frac{\partial BAC}{\partial A} = B^T C^T \qquad \frac{\partial ABA^T}{\partial A} = A[B + B^T] \tag{A54}$$

Now, solving for $\mathbf{K}_k$ in Equation (A53) gives

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{x},k}\mathbf{H}_k^T + \mathbf{R}_k + \boldsymbol{\Sigma}_{\mathbf{y},k}]^{-1} \tag{A55}$$

where the matrix term with inverse, $[\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{x},k}\mathbf{H}_k^T + \mathbf{R}_k + \boldsymbol{\Sigma}_{\mathbf{y},k}]$, is the covariance of the innovation filter, $\mathbf{P}_{\mathbf{zz},k}$.

Finally, substituting Equation (A55) into Equation (A52) yields,

$$\begin{aligned}
\mathbf{P}_k^{+} &= \mathbf{P}_k^{-} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^{-} - \mathbf{P}_k^{-}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k[\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{x},k}\mathbf{H}_k^T + \mathbf{R}_k + \boldsymbol{\Sigma}_{\mathbf{y},k}]^{-1}\mathbf{K}_k^T \\
&= \mathbf{P}_k^{-} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^{-} \\
\mathbf{P}_k^{+} &= [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^{-}
\end{aligned} \tag{A56}$$

Equation (A56) is the update equation for error covariance matrix with the optimal gain obtained in Equation (A55).

Appendix B.1.3. Key Observations

The linear quantized discrete-time Kalman filter (QDKF) closely resembles the standard Kalman filter equations. The round-off errors caused by state, measurement, and input quantization do not alter the filter structure. However, the round-off error variances

become additive to the covariances in the propagation and the gain equations as shown in Table A2. It can be noticed from Equation (A55), the Kalman gain augments the round-off error covariances as weighting factors, reducing the applied gain and thereby amplifying covariance updates. This weighting in Kalman gain can be interpreted as optimally weighing the innovation into the state updates. Another observation is that if the model is receiving quantized input, these errors percolate into the covariance propagation step (Equation (A56)) which is a deviation from standard Kalman filter equations where the forcing input is typically assumed to be unaffected by noise. If there is no forcing input in the model dynamics, this input error covariance is disregarded. The resulting QDKF algorithm serves as an optimal state estimator, accommodating quantized states, measurements, and inputs while considering uncertainties in state and measurement evolution.

Table A2. Quantized discrete-time Kalman filter algorithm.

System Model	$\mathbf{x}_{k+1} = \mathbf{\Phi}_k(\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{\Gamma}_k(\mathbf{u}_k + \boldsymbol{\epsilon}_{\mathbf{u},k}) + \gamma_k \mathbf{w}_k$
Measurement Model	$\mathbf{y}_k = \mathbf{H}_k(\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}$
Initialize	$\hat{\mathbf{x}}_0^+ = \mathbb{E}[\mathbf{x}_0]$
	$\mathbf{P}_0^+ = \mathbb{E}[(\hat{\mathbf{x}}_0 - \mathbf{x}_0)(\hat{\mathbf{x}}_0 - \mathbf{x}_0)^T]$
State propagation	$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_k^+ + \mathbf{\Gamma}_k \mathbf{u}_k$
Covariance propagation	$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k(\mathbf{P}_k^+ + \mathbf{\Sigma}_{\mathbf{x},k})\mathbf{\Phi}_k^T + \mathbf{\Gamma}_k \mathbf{\Sigma}_{\mathbf{u},k} \mathbf{\Gamma}_k^T + \gamma_k \mathbf{Q}_k \gamma_k^T$
Kalman gain	$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{H}_k \mathbf{\Sigma}_{\mathbf{x},k} \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{\Sigma}_{\mathbf{y},k}]^{-1}$
State update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$
Covariance update	$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k] \mathbf{P}_k^-$

Appendix C. Quantized Square-Root Kalman Filter (QSRKF)

Theoretically, the discrete-time Kalman filter equations are adequate to achieve state estimates with minimum mean square error. For onboard finite-precision implementation, the Kalman filter equations are prone to numerical instability. Due to round-off errors, the propagation and update equations were found to compromise the symmetry and positive-definiteness of the covariance matrix often leading to negative variances in the estimation errors [2,4]. This degradation in the Kalman filter performance is addressed through numerically robust square-root Kalman filters. In fixed-point realization, the square-root Kalman filter helps reduce the dynamic range and mitigate the round-off errors by using matrix square-root operations. The square-root Kalman filter (SRKF) operates by computing the covariance propagation and update expressions in terms of square-root factors of the *a priori* and *a posteriori* covariance matrices. This requires taking matrix square-root of the state covariance matrix.

Appendix C.1. Square-Root of State Covariance Matrix

The square-root of the state covariance matrix $\mathbf{P}_k$ is given by $\mathbf{S}_k$ and defined as:

$$\mathbf{P}_k = \mathbf{S}_k \mathbf{S}_k^T \tag{A57}$$

where the square-root factor $\mathbf{S}_k$ can be computed using Cholesky factorization of the symmetric positive-definite $\mathbf{P}_k$. Alternately, the square-root matrix can be efficiently computed using QR decomposition.

The QR algorithm decomposes a matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ into two factors: an orthogonal matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$, and an upper-triangular matrix $\mathbf{R} \in \mathbb{R}^{n \times m}$ such that $\mathbf{A} = \mathbf{QR}$, and $m \geq n$. If $\tilde{\mathbf{R}} \in \mathbb{R}^{n \times n}$ is the upper triangular portion of $\mathbf{R}$, then $\tilde{\mathbf{R}}^T$ is the Cholesky factor of $\mathbf{P}_k = \mathbf{A}\mathbf{A}^T$, i.e., $\tilde{\mathbf{R}}^T = \mathbf{S}_k$ such that $\tilde{\mathbf{R}}^T \tilde{\mathbf{R}} = \mathbf{A}\mathbf{A}^T$ [45]. Specifically, if $\tilde{\mathbf{R}} = \text{qr}\{A^T\}^T$, the $\text{qr}\{\cdot\}$ operation performs the QR decomposition and returns an upper-triangular portion of $\mathbf{R}$ only, then $\tilde{\mathbf{R}}^T$ is the lower-triangular Cholesky factor of $\mathbf{P}_k = \mathbf{A}\mathbf{A}^T$.

Appendix C.2. Quantized Square-Root Kalman Filter for State Estimation

Appendix C.2.1. Initialization

As established, the SRKF is concerned with time and measurement update using square-root of the covariance matrices. As in the QDKF formulation (See Appendix B), the QSRKF filter is initialized with an initial mean $\mathbf{x}_0^+$. However, in place of the state covariance matrix, the filter is initialized with its square-root $\mathbf{S}_0^+$ calculated once using Cholesky factorization such that:

$$\mathbf{S}_0^+ = \sqrt{\mathbb{E}[(\hat{\mathbf{x}}_0 - \mathbf{x}_0)(\hat{\mathbf{x}}_0 - \mathbf{x}_0)^T]} \tag{A58}$$

The filter description includes modeling quantization noise sources in the derivation along with the measurement and the process noises. The square-roots of the error covariances that correspond to the state, input, and measurement quantization noises (Equation (A43)), and the process, measurement noises (Equation (A38)) are calculated via Cholesky factorization. The square-root covariance matrices defined through the following notations:

$$\mathbf{\Lambda}_{\mathbf{x},k} = \sqrt{\mathbf{\Sigma}_{\mathbf{x},k}} \qquad \mathbf{\Lambda}_{\mathbf{y},k} = \sqrt{\mathbf{\Sigma}_{\mathbf{y},k}} \qquad \mathbf{\Lambda}_{\mathbf{u},k} = \sqrt{\mathbf{\Sigma}_{\mathbf{u},k}} \qquad \mathbf{S}_{\mathbf{w},k} = \sqrt{\mathbf{Q}_k} \qquad \mathbf{S}_{\mathbf{v},k} = \sqrt{\mathbf{R}_k} \tag{A59}$$

Appendix C.2.2. Filter Propagation

In the propagation step, the time update for the state estimate remains unchanged:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{\Phi}_k \hat{\mathbf{x}}_k^+ + \mathbf{\Gamma}_k \mathbf{u}_k \tag{A60}$$

The state covariance however must use only the matrix square-roots for propagation. The time-update for square-root of the state estimation error covariance $\mathbf{S}_{k+1}^-$ is derived as

$$\mathbf{P}_{k+1}^- = \mathbf{\Phi}_k \mathbf{P}_k^+ \mathbf{\Phi}_k^T + \mathbf{\Phi}_k \mathbf{\Sigma}_{\mathbf{x},k} \mathbf{\Phi}_k^T + \mathbf{\Gamma}_k \mathbf{\Sigma}_{\mathbf{u},k} \mathbf{\Gamma}_k^T + \gamma_k \mathbf{Q}_k \gamma_k^T \tag{A61}$$

$$\mathbf{S}_{k+1}^- (\mathbf{S}_{k+1}^-)^T = \mathbf{\Phi}_k \mathbf{S}_k^+ (\mathbf{S}_k^+)^T \mathbf{\Phi}_k^T + \mathbf{\Phi}_k \mathbf{\Lambda}_{\mathbf{x},k} \mathbf{\Lambda}_{\mathbf{x},k}^T \mathbf{\Phi}_k^T + \mathbf{\Gamma}_k \mathbf{\Lambda}_{\mathbf{u},k} \mathbf{\Lambda}_{\mathbf{u},k}^T \mathbf{\Gamma}_k^T + \gamma_k \mathbf{S}_{\mathbf{w},k} \mathbf{S}_{\mathbf{w},k}^T \gamma_k^T \tag{A62}$$

$$\mathbf{S}_{k+1}^- (\mathbf{S}_{k+1}^-)^T = \left[\mathbf{\Phi}_k \mathbf{S}_k^+ \mid \mathbf{\Phi}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{\Gamma}_k \mathbf{\Lambda}_{\mathbf{u},k} \mid \gamma_k \mathbf{S}_{\mathbf{w},k}\right] \left[\mathbf{\Phi}_k \mathbf{S}_k^+ \mid \mathbf{\Phi}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{\Gamma}_k \mathbf{\Lambda}_{\mathbf{u},k} \mid \gamma_k \mathbf{S}_{\mathbf{w},k}\right]^T \tag{A63}$$

$$\mathbf{S}_{k+1}^- = \mathrm{qr}\{\left[\mathbf{\Phi}_k \mathbf{S}_k^+ \mid \mathbf{\Phi}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{\Gamma}_k \mathbf{\Lambda}_{\mathbf{u},k} \mid \gamma_k \mathbf{S}_{\mathbf{w},k}\right]^T\}^T \tag{A64}$$

Appendix C.2.3. Kalman Gain

To compute the Kalman gain for the QSRKF, the expressions for the square-root of the innovation covariance is first developed. The innovation covariance $\mathbf{P}_{\mathbf{zz}}$ of the QDKF algorithm (Equation (A55)) is given as:

$$\mathbf{P}_{\mathbf{zz},k} = \mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{H}_k \mathbf{\Sigma}_{\mathbf{x},k} \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{\Sigma}_{\mathbf{y},k} \tag{A65}$$

Similar to the propagation step, the square-root factors for the innovation covariance is developed as:

$$\mathbf{S}_{\mathbf{zz},k} \mathbf{S}_{\mathbf{zz},k}^T = \mathbf{H}_k \mathbf{S}_k^- (\mathbf{S}_k^-)^T \mathbf{H}_k^T + \mathbf{H}_k \mathbf{\Lambda}_{\mathbf{x},k} \mathbf{\Lambda}_{\mathbf{x},k}^T \mathbf{H}_k^T + \mathbf{S}_{\mathbf{v},k} \mathbf{S}_{\mathbf{v},k}^T + \mathbf{\Lambda}_{\mathbf{y},k} \mathbf{\Lambda}_{\mathbf{y},k}^T \tag{A66}$$

$$\mathbf{S}_{\mathbf{zz},k} = \mathrm{qr}\{\left[\mathbf{H}_k \mathbf{S}_k^- \mid \mathbf{H}_k \mathbf{\Lambda}_{\mathbf{x},k} \mid \mathbf{S}_{\mathbf{v},k} \mid \mathbf{\Lambda}_{\mathbf{y},k}\right]^T\}^T \tag{A67}$$

The Kalman gain $\mathbf{K}_k$ of the QDKF method is known to be $\mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{P}_{\mathbf{zz},k})^{-1}$. Using the calculated square-root innovation covariance factor, the Kalman gain is computed as:

$$\mathbf{K}_k = \mathbf{S}_k^- (\mathbf{H}_k \mathbf{S}_k^-)^T (\mathbf{S}_{\mathbf{zz},k} \mathbf{S}_{\mathbf{zz},k}^T)^{-1} \tag{A68}$$

Appendix C.2.4. Filter Update

The state and the state covariance update equations are computed using the Kalman gain presented in Equation (A68). Starting from the symmetric version of the covariance update equation (Equation (50)), the square-root factored form of the covariance update, $\mathbf{S}_k^+$, is computed as:

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^-[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \mathbf{K}_k[\mathbf{H}_k\boldsymbol{\Sigma}_{\mathbf{x},k}\mathbf{H}_k^T + \mathbf{R}_k + \boldsymbol{\Sigma}_{\mathbf{y},k}]\mathbf{K}_k^T \tag{A69}$$

$$\begin{aligned}\mathbf{S}_k^+(\mathbf{S}_k^+)^T = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{S}_k^-(\mathbf{S}_k^-)^T[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]^T + \\ \mathbf{K}_k[\mathbf{H}_k\boldsymbol{\Lambda}_{\mathbf{x},k}\boldsymbol{\Lambda}_{\mathbf{x},k}^T\mathbf{H}_k^T + \mathbf{S}_{\mathbf{v},k}\mathbf{S}_{\mathbf{v},k}^T + \boldsymbol{\Lambda}_{\mathbf{y},k}\boldsymbol{\Lambda}_{\mathbf{y},k}^T]\mathbf{K}_k^T\end{aligned} \tag{A70}$$

$$\mathbf{S}_k^+ = \mathrm{qr}\Big\{\Big[[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{S}_k^- \mid \mathbf{K}_k\big[\mathbf{H}_k\boldsymbol{\Lambda}_{\mathbf{x},k} \mid \mathbf{S}_{\mathbf{v},k} \mid \boldsymbol{\Lambda}_{\mathbf{y},k}\big]\Big]^T\Big\}^T \tag{A71}$$

Table A3. Quantized square-root Kalman filter algorithm.

System Model	$\mathbf{x}_{k+1} = \boldsymbol{\Phi}_k(\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \boldsymbol{\Gamma}_k(\mathbf{u}_k + \boldsymbol{\epsilon}_{\mathbf{u},k}) + \gamma_k\mathbf{w}_k$
Measurement Model	$\mathbf{y}_k = \mathbf{H}_k(\mathbf{x}_k + \boldsymbol{\epsilon}_{\mathbf{x},k}) + \mathbf{v}_k + \boldsymbol{\epsilon}_{\mathbf{y},k}$
Initialize	$\hat{\mathbf{x}}_0^+ = \mathbb{E}[\mathbf{x}_0]$ $\mathbf{S}_0^+ = \sqrt{\mathbb{E}[(\hat{\mathbf{x}}_0 - \mathbf{x}_0)(\hat{\mathbf{x}}_0 - \mathbf{x}_0)^T]}$
State propagation	$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\Phi}_k\hat{\mathbf{x}}_k^+ + \boldsymbol{\Gamma}_k\mathbf{u}_k$
Square-root covariance propagation	$\mathbf{S}_{k+1}^- = \mathrm{qr}\{\big[\boldsymbol{\Phi}_k\mathbf{S}_k^+ \mid \boldsymbol{\Phi}_k\boldsymbol{\Lambda}_{\mathbf{x},k} \mid \boldsymbol{\Gamma}_k\boldsymbol{\Lambda}_{\mathbf{u},k} \mid \gamma_k\mathbf{S}_{\mathbf{w},k}\big]^T\}^T$
Innovation Covariance	$\mathbf{S}_{\mathbf{zz},k} = \mathrm{qr}\{\big[\mathbf{H}_k\mathbf{S}_k^- \mid \mathbf{H}_k\boldsymbol{\Lambda}_{\mathbf{x},k} \mid \mathbf{S}_{\mathbf{v},k} \mid \boldsymbol{\Lambda}_{\mathbf{y},k}\big]^T\}^T$
Kalman gain	$\mathbf{K}_k = \mathbf{S}_k^-(\mathbf{H}_k\mathbf{S}_k^-)^T(\mathbf{S}_{\mathbf{zz},k}\mathbf{S}_{\mathbf{zz},k}^T)^{-1}$
State update	$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$
Square-root covariance update	$\mathbf{S}_k^+ = \mathrm{qr}\Big\{\Big[[\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{S}_k^- \mid \mathbf{K}_k\big[\mathbf{H}_k\boldsymbol{\Lambda}_{\mathbf{x},k} \mid \mathbf{S}_{\mathbf{v},k} \mid \boldsymbol{\Lambda}_{\mathbf{y},k}\big]\Big]^T\Big\}^T$

References

1. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
2. Grewal, M.S.; Andrews, A.P. Applications of Kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control. Syst. Mag.* **2010**, *30*, 69–78. [CrossRef]
3. Hall, E.C. *Journey to the Moon: The History of the Apollo Guidance Computer*; AIAA: Reston, VA, USA, 1996.
4. Kaminski, P.; Bryson, A.; Schmidt, S. Discrete square root filtering: A survey of current techniques. *IEEE Trans. Autom. Control* **1971**, *16*, 727–736. [CrossRef]
5. Morf, M.; Kailath, T. Square-root algorithms for least-squares estimation. *IEEE Trans. Autom. Control* **1975**, *20*, 487–497. [CrossRef]
6. Gaston, F.; Irwin, G. Systolic approach to square root information Kalman filtering. *Int. J. Control* **1989**, *50*, 225–248. [CrossRef]
7. Chin, T.M.; Karl, W.C.; Willsky, A.S. A distributed and iterative method for square root filtering in space-time estimation. *Automatica* **1995**, *31*, 67–82. [CrossRef]
8. Lee, C.R.; Salcic, Z. High-performance FPGA-based implementation of Kalman filter. *Microprocess. Microsystems* **1997**, *21*, 257–265. [CrossRef]
9. Dutt, R.; Acharyya, A. Low-Complexity Square-Root Unscented Kalman Filter Design Methodology. *Circuits Syst. Signal Process.* **2023**, *42*, 6900–6928. [CrossRef]
10. Liu, K.; Skelton, R. Optimal controllers for finite wordlength implementation. In Proceedings of the 1990 American Control Conference, San Diego, CA, USA, 23–25 May 1990; pp. 1935–1940.
11. Wong, W.S.; Brockett, R.W. Systems with finite communication bandwidth constraints. I. State estimation problems. *IEEE Trans. Autom. Control* **1997**, *42*, 1294–1299. [CrossRef]
12. You, K.; Xie, L.; Sun, S.; Xiao, W. Quantized filtering of linear stochastic systems. *Trans. Inst. Meas. Control* **2011**, *33*, 683–698. [CrossRef]
13. Ribeiro, A.; Giannakis, G.B.; Roumeliotis, S.I. SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations. *IEEE Trans. Signal Process.* **2006**, *54*, 4782–4795. [CrossRef]
14. Hong, S.; Bolic, M.; Djuric, P.M. An efficient fixed-point implementation of residual resampling scheme for high-speed particle filters. *IEEE Signal Process. Lett.* **2004**, *11*, 482–485. [CrossRef]
15. Soh, J. A Scalable, Portable, FPGA-Based Implementation of the Unscented Kalman Filter. Ph.D. Thesis, The University of Sydney, Sydney, Australia, 2017.
16. Babu, P.; Parthasarathy, E. FPGA implementation of multi-dimensional Kalman filter for object tracking and motion detection. *Eng. Sci. Technol. Int. J.* **2022**, *33*, 101084. [CrossRef]

17. Guzmán Cervantes, F.; Kumanchik, L.; Pratt, J.; Taylor, J.M. High sensitivity optomechanical reference accelerometer over 10 kHz. *Appl. Phys. Lett.* **2014**, *104*, 221111. [CrossRef]
18. Cervantes, F.G.; Flatscher, R.; Gerardi, D.; Burkhardt, J.; Gerndt, R.; Nofrarias, M.; Reiche, J.; Heinzel, G.; Danzmann, K.; Boté, L.G.; et al. LISA technology package flight hardware test campaign. In Proceedings of the ASP Conference Series, Seattle, WA, USA, 7–12 July 2013; Volume 467, pp. 141–150.
19. Kornfeld, R.P.; Arnold, B.W.; Gross, M.A.; Dahya, N.T.; Klipstein, W.M.; Gath, P.F.; Bettadpur, S. GRACE-FO: The Gravity Recovery and Climate Experiment Follow-On Mission. *J. Spacecr. Rocket.* **2019**, *56*, 931–951. [CrossRef]
20. Amaro-Seoane, P.; Audley, H.; Babak, S.; Baker, J.; Barausse, E.; Bender, P.; Berti, E.; Binetruy, P.; Born, M.; Bortoluzzi, D.; et al. Laser interferometer space antenna. *arXiv* **2017**, arXiv:1702.00786.
21. Abbott, B.; Abbott, R.; Adhikari, R.; Ajith, P.; Allen, B.; Allen, G.; Amin, R.; Anderson, S.; Anderson, W.; Arain, M.; et al. LIGO: The laser interferometer gravitational-wave observatory. *Rep. Prog. Phys.* **2009**, *72*, 076901. [CrossRef]
22. Iturbe, X.; Keymeulen, D.; Ozer, E.; Yiu, P.; Berisford, D.; Hand, K.; Carlson, R. An integrated SoC for science data processing in next-generation space flight instruments avionics. In Proceedings of the 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), Daejeon, Republic of Korea, 5–7 October 2015; pp. 134–141.
23. Ramchander Rao, B. Hardware Implementation of Navigation Filters for Automation of Dynamical Systems. Master's Thesis, Texas A&M University, College Station, TX, USA, 2021.
24. Farrenkopf, R. Analytic steady-state accuracy solutions for two common spacecraft attitude estimators. *J. Guid. Control* **1978**, *1*, 282–284. [CrossRef]
25. Oppenheim, A.V.; Schafer, R.W. *Digital Signal Processing*; Research Supported by the Massachusetts Institute of Technology, Bell Telephone Laboratories, and Guggenheim Foundation; Prentice-Hall, Inc.: Saddle River, NJ, USA, 1975; 598p.
26. Mullis, C.; Roberts, R. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Trans. Circuits Syst.* **1976**, *23*, 551–562. [CrossRef]
27. Kelly, P.; Majji, M.; Guzmán, F. Estimation and Error Analysis for Optomechanical Inertial Sensors. *Sensors* **2021**, *21*, 6101. [CrossRef]
28. Hines, A.; Richardson, L.; Wisniewski, H.; Guzman, F. Optomechanical inertial sensors. *Appl. Opt.* **2020**, *59*, G167–G174. [CrossRef] [PubMed]
29. Rasras, M.; Elfadel, I.A.M.; Ngo, H.D. *MEMS Accelerometers*; MDPI: Basel, Switzerland, 2019.
30. Reschovsky, B.J.; Long, D.A.; Zhou, F.; Bao, Y.; Allen, R.A.; LeBrun, T.W.; Gorman, J.J. Intrinsically accurate sensing with an optomechanical accelerometer. *Opt. Express* **2022**, *30*, 19510–19523. [CrossRef] [PubMed]
31. Gerberding, O.; Cervantes, F.G.; Melcher, J.; Pratt, J.R.; Taylor, J.M. Optomechanical reference accelerometer. *Metrologia* **2015**, *52*, 654. [CrossRef]
32. Hwang, S. Roundoff noise in state-space digital filtering: A general analysis. *IEEE Trans. Acoust. Speech, Signal Process.* **1976**, *24*, 256–262. [CrossRef]
33. Williamson, D.; Kadiman, K. Optimal finite wordlength linear quadratic regulation. *IEEE Trans. Autom. Control* **1989**, *34*, 1218–1228. [CrossRef]
34. Sripad, A.; Snyder, D. A necessary and sufficient condition for quantization errors to be uniform and white. *IEEE Trans. Acoust. Speech Signal Process.* **1977**, *25*, 442–448. [CrossRef]
35. Crassidis, J.L.; Junkins, J.L. *Optimal Estimation of Dynamic Systems*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2004.
36. Elmenreich, W. Sensor Fusion in Time-Triggered Systems. Ph.D. Thesis, The Pennsylvania State University, University Park, PA, USA, 2002.
37. Bierman, G.J. *Factorization Methods for Discrete Sequential Estimation*; Courier Corporation: Chelmsford, MA, USA, 2006.
38. Wisniewski, H.; Richardson, L.; Hines, A.; Laurain, A.; Guzmán, F. Optomechanical lasers for inertial sensing. *JOSA A* **2020**, *37*, B87–B92. [CrossRef]
39. Kung, H.T. Why systolic architectures? *Computer* **1982**, *15*, 37–46. [CrossRef]
40. Golub, G.H.; Van Loan, C.F. *Matrix Computations*; JHU Press: Baltimore, MD, USA, 2013.
41. Gentleman, W.M. Least squares computations by Givens transformations without square roots. *IMA J. Appl. Math.* **1973**, *12*, 329–336. [CrossRef]
42. Bhaskara, R.R.; Sung, K.; Majji, M. An FPGA framework for Interferometric Vision-Based Navigation (iVisNav). In Proceedings of the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 18–22 September 2022; pp. 1–7.
43. Anderson, B.D. Exponential data weighting in the Kalman-Bucy filter. *Inf. Sci.* **1973**, *5*, 217–230. [CrossRef]
44. Jazwinski, A.H. *Stochastic Processes and Filtering Theory*; Courier Corporation: Chelmsford, MA, USA, 2007.
45. Van Der Merwe, R.; Wan, E.A. The square-root unscented Kalman filter for state and parameter-estimation. In Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings (Cat. No. 01CH37221), Salt Lake City, UT, USA, 7–11 May 2001; Volume 6, pp. 3461–3464.