ARL-TR-9165 • MAR 2021

A Modular Approach to Kalman Filter Design and Analysis

by James Maley

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-9165 ● MAR 2021

DEVCOM
ARMY RESEARCH
LABORATORY

A Modular Approach to Kalman Filter Design and Analysis

by James Maley
Weapons and Materials Research Directorate, DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE	***Form Approved*** ***OMB No. 0704-0188***

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* March 2021	**2. REPORT TYPE** Technical Report	**3. DATES COVERED (From - To)** September 2018–Present

4. TITLE AND SUBTITLE A Modular Approach to Kalman Filter Design and Analysis	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) James Maley	**5d. PROJECT NUMBER** AH80
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) DEVCOM Army Research Laboratory ATTN: FCDD-RLW-LF Aberdeen Proving Ground, MD 21005-5066	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-9165
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

primary author's email: <james.m.maley2.civ@mail.mil>.

14. ABSTRACT

Extended Kalman filters are important in navigation, guidance, and parameter estimation problems, and typically combine information from several different systems. This report discusses using error-states to model, linearize, and partition individual component systems. The system models are later combined into filters, which maintain the partitioning of the individual systems. This partitioning reduces the complexity of the covariance propagation and update operations. An example aided inertial navigation system is studied under two trajectories; tumble-test calibration and guided munition flight. Linearized error budgets for the munition trajectory were used to determine which sensor calibration errors it made the most sense to actively estimate, and which sensor errors contributed the most to the error in the optimal case where everything was estimated.

15. SUBJECT TERMS

Kalman filtering, State Estimation, Navigation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** James M Maley
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UU	60	**19b. TELEPHONE NUMBER (Include area code)** 410-306-0814

Contents

List of Figures

List of Tables

1. Introduction

Kalman filtering and extended Kalman filtering are well-known techniques for estimating the states of stochastic systems of differential equations. They are used extensively in navigation systems, attitude estimators, parameter estimation, and many other applications. Building the Kalman filter consists of modeling the processes and measurements involved using state-space methods, and building an extended Kalman filter requires knowledge of how to properly linearize a nonlinear system. As such, an Extended Kalman Filter (EKF) can be challenging to build, tune, analyze, and implement. In general, the more accurately the system is modeled, the better the state estimator will perform.

The purpose of this report is to present a modular approach to building Kalman filters that model the system as completely as possible, yet still run efficiently through matrix partitioning, and how to use covariance analysis as a design and analysis tool for these filters. The approach is built upon the EKF and Schmidt Kalman filter basics presented in state estimation references such as Simon,[1] Brown and Hwang,[2] Maybeck,[3] or Gelb.[4] Since the methods documented here will be repeatedly used in other global positioning system (GPS)-denied navigation projects, it would be beneficial to document the underlying approach in one place.

The EKF presented is implemented as an error-state Kalman filter. Using error states is a commonly used approach in navigation filters because they allow for nonlinear update rules that allow rotations to stay within the $SO(3)$ (or unit quaternion) group.[5] Though more ad-hoc than matrix Lie group-based approaches (e.g., see Solà[6]), error-state mechanizations often provide Jacobians that are equivalent up to a first order approximation. In addition, some of the "exactness" achieved using Lie approaches vanishes when sensor calibration errors are present in the model. A generic error state convention is discussed that is capable of representing any systems used in navigation, but also is easy to use with symbolic math software such as MathWorks, MATLAB's Symbolic Math Toolbox,[7] in order to compute Jacobians and propagation matrices. Partitioning the states into $dynamic$ states that change with time and $static$ states that remain constant reduces the complexity of the propagation step. Partitioning the states into $active$ states that are estimated by the Kalman filter, and $consider$ states that are not estimated, but have their uncertainty accounted for when computing the Kalman gain, can reduce the complexity of the update step. Each dynamic system model is partitioned into the four resulting permutations of state categories. Multiple partitioned dynamic system models are then combined to form filters.

An example is provided at the end to demonstrate the benefits of the matrix partitioning proposed here. A navigation system is constructed that combines a black-box GPS receiver with an inertial measurement unit (IMU) and magnetometer to estimate the attitude, position, velocity, and various sensor calibration errors. The GPS, IMU, and magnetometer are all modeled using separate dynamic systems. Each dynamic system is modeled in three different ways: one in which a high fidelity error model is used with a mixture of active and consider states, one in which a simplified model is used with covariance inflation, and one in which a high fidelity model is used without consider states. Three Kalman filters are constructed from the dynamic system models. The filters are then tested against a tumble-test trajectory and a guided munition trajectory. From the simulation results, the simplified filter is shown to be both less accurate and less consistent, but faster. Additional tuning could probably improve the performance of the simplified model filter, but the high fidelity models were tuned directly from the sensor specifications and did not need a manual tuning step. The high fidelity filter with consider states is shown to be faster but less accurate than the one without consider states.

The example is extended by designing a slightly different filter based on the error budget of the consider state filter. The error budget illuminates which errors would improve the navigation performance the most if they were modeled as active states instead of consider states. The new filter with these modeling changes performs almost identically to the optimal filter. Complexity analysis was performed on the different filters, and an error budget was performed on the optimal filter to determine if there were any sensor-quality limiting factors. The presented navigation filters demonstrate the ability to reduce computational complexity, yet maintain near-optimal state estimation, and offer the filter designer a clear methodology to trade accuracy versus speed using consider states.

2. Error State Systems

Error state Kalman filters differ from normal EKFs in that they do not directly estimate the state. Instead, it is assumed that the true state $\boldsymbol{x}$ can be composed of the estimated state $\hat{\boldsymbol{x}}$ and the error state $\tilde{\mathrm{x}}$ with

$$\boldsymbol{x} = \tilde{\mathrm{x}} \oplus \hat{\boldsymbol{x}} \tag{1}$$

The error state Kalman filter computes Jacobians with respect to $\tilde{\mathrm{x}}$, and updates an estimate of the error states $\hat{\tilde{\mathrm{x}}}$ in the innovation step. The error state is the only quantity that has to be expressible as a vector. The actual states and their estimates can be sets of objects belonging to different mathematical groups such as rotation matrices in addition to real vectors, and $\oplus$ would be the “product” operator in group theory language. Because $\oplus$ applies a correction to the estimated states, it will be referred to here as the “correction operator”.

The basic module of the Kalman filter is a system (of stochastic differential equations). It is up to the designer to determine which errors effect a sensor or process. Consider the two (greatly simplified) systems described in Table 1. The attitude system uses a quaternion as the state, but a small-angle vector as the error state. The quaternion evolves through time as a function of the angular rates, which are measured by a gyroscope. The gyroscope outputs are considered exogeneous inputs to the system. Likewise, noise on the gyroscope causes the error-states to change in time. [*] The magnetometer system only includes a magnetometer bias as the state, and an error in this bias as the error state, which are both constant. The magnetometer produces an output: the measured magnetic field. This output is a function not only of the magnetometer states but the attitude states.

The propagation and measurement steps are described in more detail in the following subsections. The goal is to be able to define the errors of each system independently, and then to combine multiple systems into a Kalman filter.

[*] Attitude error propagation has been studied extensively. We refer the reader to Trawny’s technical report[8] (JPL quaternion convention used here) or Maley’s technical report[9] (Hamiltonian quaternion convention) for details.

Table 1 Simple conceptual example

System	States	Error States	Inputs/External States	Outputs
attitude	quaternion	small angles	gyroscope outputs	none
magnetometer	bias	bias error	quaternion	measurement magnetic field

2.1 Propagation

The vast majority of things that need to be modeled in a Kalman filter can be represented by states $\boldsymbol{x}$ that propagate according to

$$\dot{\boldsymbol{x}} = f\left(\boldsymbol{x}, \mathbf{u}, \mathbf{w}\right), \tag{2}$$

where $\mathbf{u}$ is a vector of exogeneous inputs and $\mathbf{w}$ is a vector of uncorrelated white noise with a Power Spectral Density (PSD) of 1. It is assumed that $\mathbf{u}$ is known. The state estimates will propagate by integrating $\dot{\hat{\boldsymbol{x}}} = f\left(\hat{\boldsymbol{x}}, \mathbf{u}\right)$ from time t_k to time t_{k+1} using some numerical integration scheme (Euler, trapezoidal, Simpson's rule etc.). In order to propagate the error-state covariance, the error-state dynamics must be used. The error-state dynamics propagate according to

$$\dot{\tilde{\mathbf{x}}} = \tilde{f}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \mathbf{w}\right). \tag{3}$$

The function $\tilde{f}$ can be obtained from Eq. 1 and Eq. 2 and treating products of error states as negligible. For example (similar to the derivation in Titterton[10] just with a different angle error convention), consider a simple system consisting of a rotation matrix $\mathbf{R}$ and a velocity vector $\mathbf{v}$ with:

- state estimates $\hat{\boldsymbol{x}} = \left\{\hat{\mathbf{R}}, \quad \hat{\mathbf{v}}\right\}$
- error state vector $\tilde{\mathbf{x}} = \begin{bmatrix}\tilde{\boldsymbol{\theta}}^\top & \tilde{\mathbf{v}}^\top\end{bmatrix}^\top$

- correction operator

$$\begin{aligned} \boldsymbol{x} &= \Big\{\mathbf{R}, \quad \mathbf{v}\Big\} = \tilde{\mathbf{x}} \otimes \hat{\boldsymbol{x}} \\ &= \Big\{\exp\Big(-\Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\Big)\,\hat{\mathbf{R}}, \quad \tilde{\mathbf{v}} + \hat{\mathbf{v}}\Big\} \\ &\approx \Big\{\Big(\mathbf{I} - \Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\Big)\,\hat{\mathbf{R}}, \quad \tilde{\mathbf{v}} + \hat{\mathbf{v}}\Big\} \end{aligned}$$

- and dynamics $\dot{\boldsymbol{x}} = \Big\{\dot{\mathbf{R}} = -\lfloor\boldsymbol{\omega}\times\rfloor\,\mathbf{R}, \quad \dot{\mathbf{v}} = \mathbf{R}^\top\mathbf{a} + \mathbf{g}\Big\}$
  where $\boldsymbol{\omega}$, $\mathbf{a}$, and $\mathbf{g}$ are perfect gyroscope, accelerometer, and gravity values.

Because inertial outputs are imperfect, $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}$, and $\mathbf{a} = \hat{\mathbf{a}} + \tilde{\mathbf{a}}$. By applying the correction function operator to the dynamics, one obtains for the attitude errors

$$\begin{aligned} \dot{\mathbf{R}} &= -\lfloor\boldsymbol{\omega}\times\rfloor\,\mathbf{R} \\ \frac{d}{dt}\Big(\Big(\mathbf{I} - \Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\Big)\,\hat{\mathbf{R}}\Big) &\approx -\left(\lfloor\hat{\boldsymbol{\omega}}\times\rfloor + \lfloor\tilde{\boldsymbol{\omega}}\times\rfloor\right)\Big(\mathbf{I} - \Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\Big)\,\hat{\mathbf{R}} \\ -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\hat{\mathbf{R}} + \Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\,\lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\hat{\mathbf{R}} - \Big\lfloor\dot{\tilde{\boldsymbol{\theta}}}\times\Big\rfloor\,\hat{\mathbf{R}} &\approx -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\hat{\mathbf{R}} - \lfloor\tilde{\boldsymbol{\omega}}\times\rfloor\,\hat{\mathbf{R}} + \lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\,\hat{\mathbf{R}} \\ \Big\lfloor\dot{\tilde{\boldsymbol{\theta}}}\times\Big\rfloor &\approx \Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\,\lfloor\hat{\boldsymbol{\omega}}\times\rfloor - \lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor + \lfloor\tilde{\boldsymbol{\omega}}\times\rfloor \\ \dot{\tilde{\boldsymbol{\theta}}} &\approx -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\omega}} \end{aligned}$$

and for the velocity errors

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{R}^\top\mathbf{a} + \mathbf{g} \\ \dot{\tilde{\mathbf{v}}} + \dot{\hat{\mathbf{v}}} &\approx \hat{\mathbf{R}}^\top\Big(\mathbf{I} + \Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\Big)\left(\hat{\mathbf{a}} + \hat{\mathbf{a}}\right) + \mathbf{g} \\ \dot{\tilde{\mathbf{v}}} + \dot{\hat{\mathbf{v}}} &\approx \dot{\hat{\mathbf{v}}} + \hat{\mathbf{R}}^\top\Big\lfloor\tilde{\boldsymbol{\theta}}\times\Big\rfloor\,\hat{\mathbf{a}} + \hat{\mathbf{R}}^\top\tilde{\mathbf{a}} \\ \dot{\tilde{\mathbf{v}}} &= -\hat{\mathbf{R}}^\top\lfloor\hat{\mathbf{a}}\times\rfloor\,\tilde{\boldsymbol{\theta}} + \hat{\mathbf{R}}^\top\tilde{\mathbf{a}} \end{aligned}$$

This leads to the error dynamics function

$$\underbrace{\begin{bmatrix} \dot{\tilde{\boldsymbol{\theta}}} \\ \dot{\tilde{\mathbf{v}}} \end{bmatrix}}_{\dot{\tilde{\mathbf{x}}}} = \underbrace{\begin{bmatrix} -\lfloor\hat{\boldsymbol{\omega}}\times\rfloor\,\tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\omega}} \\ -\hat{\mathbf{R}}^\top\lfloor\hat{\mathbf{a}}\times\rfloor\,\tilde{\boldsymbol{\theta}} + \hat{\mathbf{R}}^\top\tilde{\mathbf{a}} \end{bmatrix}}_{\tilde{f}(\hat{\boldsymbol{x}},\tilde{\mathbf{x}},\mathbf{u},\mathbf{w})} \tag{4}$$

There is significant flexibility in how the inertial outputs are modeled, which is further discussed in Section 5. In this small example, since only the attitude and velocity were defined, we will assume the sensor estimates are the raw sensor outputs, and the error is noise, which would make $\mathbf{u} = \begin{bmatrix}\hat{\boldsymbol{\omega}}^\top & \hat{\mathbf{a}}^\top\end{bmatrix}^\top = \begin{bmatrix}\breve{\boldsymbol{\omega}}^\top & \breve{\mathbf{a}}^\top\end{bmatrix}^\top$, $\begin{bmatrix}\tilde{\boldsymbol{\omega}}^\top & \tilde{\mathbf{a}}^\top\end{bmatrix}^\top = \begin{bmatrix}\sigma_\omega \mathbf{w}_\omega^\top & \sigma_a \mathbf{w}_a^\top\end{bmatrix}^\top$, and $\mathbf{w} = \begin{bmatrix}\mathbf{w}_\omega^\top & \mathbf{w}_a^\top\end{bmatrix}$, where σ_ω and σ_a are the sensor noise standard deviations.

Once it is defined, it is easy to work with $\tilde{f}$ to generate the covariance propagation matrices for the system. A first order Taylor series expansion around the nominal conditions of $\tilde{\mathbf{x}} = \mathbf{0}$ and $\mathbf{w} = \mathbf{0}$ leads to

$$\dot{\tilde{\mathbf{x}}} \approx \tilde{f}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \mathbf{w}\right)\Bigg|_{\tilde{\mathbf{x}}=0,\mathbf{w}=0} + \mathbf{A}\left(\tilde{\mathbf{x}} - \mathbf{0}\right) + \mathbf{G}\left(\mathbf{w} - \mathbf{0}\right), \tag{5}$$

where $\mathbf{A}$ and $\mathbf{G}$ are defined as

$$\begin{aligned}\mathbf{A} &= \lceil \tilde{f}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \mathbf{w}\right)\rceil_{\tilde{\mathbf{x}}}\Bigg|_{\tilde{\mathbf{x}}=0,\mathbf{w}=0} \\ \mathbf{G} &= \lceil \tilde{f}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \mathbf{w}\right)\rceil_{\mathbf{w}}\Bigg|_{\tilde{\mathbf{x}}=0,\mathbf{w}=0}.\end{aligned}$$

Because of how it is defined, $\tilde{f}$ evaluated with $\tilde{\mathbf{x}} = \mathbf{0}$ and $\mathbf{w} = \mathbf{0}$ is $\mathbf{0}$. This is consistent with the expected error in Eq. 2 being equal to $\mathbf{0}$. The initial errors must be small (e.g., less than approx. $10°$ for rotation errors) in order for the Taylor series in Eq. 5 (and everything that follows) to be valid. So the error state propagation simplifies to:

$$\dot{\tilde{\mathbf{x}}} \approx \mathbf{A}\tilde{\mathbf{x}} + \mathbf{G}\mathbf{w} \tag{6}$$

Although it is possible to integrate the continuous covariance propagation equation (the Sylvester equation)[1] the continuous time error dynamics are usually discretized in order to propagate the error state covariance according to

$$\mathbf{P}_k = \mathbf{\Phi}_{k|k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k|k-1}^\top + \mathbf{Q}_{k|k-1}.$$

The state transition matrix defined as a function of the dummy time variable τ is:

$$\mathbf{\Phi}_{\tau|t} = e^{\mathbf{A}(\tau-t)} = \sum_{i=0}^{\infty}\frac{1}{i!}\left(\mathbf{A}(\tau-t)\right)^i = \mathbf{I} + \mathbf{A}(\tau-t) + \frac{1}{2}\mathbf{A}^2(\tau-t)^2\ldots. \tag{7}$$

The covariance of the process noise in between t_{k-1} and time t_k is given by:

$$\mathbf{Q}_{k|k-1} = \int_{t_{k-1}}^{t_k} \mathbf{\Phi}_{\tau|t_{k-1}} \mathbf{G}\mathbf{G}^\top \mathbf{\Phi}_{\tau|t_{k-1}}^\top d\tau \tag{8}$$

The shorthand $\mathbf{\Phi}_{k|k-1} = \mathbf{\Phi}_{t_k|t_{k-1}}$, and $\mathbf{Q}_{k|k-1} = \mathbf{Q}_{t_k|t_{k-1}}$ will be used to represent the state transition and covariance matrices in between time t_{k-1} and time t_k. Note that $\mathbf{A}$ and $\mathbf{G}$ are functions of the current state estimate $\hat{\boldsymbol{x}}$ and possibly $\mathbf{u}$. This means that $\mathbf{\Phi}_{k|k-1}$ and sometimes $\mathbf{Q}_{k|k-1}$ will also be functions of the state estimate and inputs. As an approximation, it is usually assumed that the state estimates and inputs are approximately constant over the time step. Because the state prediction has been integrated, the state vector is interpolated halfway between $\hat{\boldsymbol{x}}_{k-1}$ and $\hat{\boldsymbol{x}}_k$ (care must be taken doing this with rotations).

2.2 Outputs

Systems usually have one or more outputs of the form:

$$\boldsymbol{z} = h(\boldsymbol{x}, \mathbf{u}, \boldsymbol{e}, \mathbf{v})$$

Where the state vector and inputs are joined by external states from other systems $\boldsymbol{e}$ and measurement noise $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The estimated outputs are functions of the estimated states:

$$\hat{\boldsymbol{z}} = h(\hat{\boldsymbol{x}}, \hat{\mathbf{u}}, \hat{\boldsymbol{e}})$$

In general, the outputs, estimated outputs, and output errors are related by nonlinear operators $\oplus$ and $\ominus$ such that

$$\begin{aligned}
\boldsymbol{z} &= \tilde{\mathbf{z}} \oplus \hat{\boldsymbol{z}} \\
\tilde{\mathbf{z}} &= \boldsymbol{z} \ominus \hat{\boldsymbol{z}} = \tilde{h}(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \hat{\boldsymbol{e}}, \tilde{\mathbf{e}}, \mathbf{v})
\end{aligned}$$

As in the propagation step, the measurement error function $\tilde{h}$ can be expanded about the no-error condition to obtain linear mappings for the error states to the residuals

$$\tilde{\mathbf{z}} \approx \mathbf{H}\tilde{\mathbf{x}} + \mathbf{E}\tilde{\mathbf{e}} + \mathbf{L}\mathbf{v}, \tag{9}$$

were

$$\begin{aligned}
\mathbf{H} &= \lceil \tilde{h}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \hat{\boldsymbol{e}}, \tilde{\mathbf{e}}, \mathbf{v}\right) \rceil_{\tilde{\mathbf{x}}} \Big|_{\tilde{\mathbf{x}}=0, \tilde{\mathbf{e}}=0, \mathbf{v}=0} \\
\mathbf{E} &= \lceil \tilde{h}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \hat{\boldsymbol{e}}, \tilde{\mathbf{e}}, \mathbf{v}\right) \rceil_{\tilde{\mathbf{e}}} \Big|_{\tilde{\mathbf{x}}=0, \tilde{\mathbf{e}}=0, \mathbf{v}=0} \\
\mathbf{L} &= \lceil \tilde{h}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \hat{\boldsymbol{e}}, \tilde{\mathbf{e}}, \mathbf{v}\right) \rceil_{\tilde{\mathbf{v}}} \Big|_{\tilde{\mathbf{x}}=0, \tilde{\mathbf{e}}=0, \mathbf{v}=0} .
\end{aligned}$$

2.3 Kalman Updates

It is the *error state* that is being estimated during the Kalman update. Once the error state is estimated, it is used to correct the states using the $\oplus$ operator, after which the error state is set to 0. To state this explicitly, suppose $\hat{\tilde{\mathbf{x}}}^{-}$ denotes the estimated error state maintained by the EKF algorithm. Here, we will use $^{-}$ and $^{+}$ superscripts to distinguish between quantities before and after performing the nonlinear update, respectively. We will temporarily use subscripts to denote how many measurements have been processed, so after processing the z measurements the estimated error state is $\hat{\tilde{\mathbf{x}}}_z^{-}$, and its covariance is $\mathbf{P}_z^{-} = \mathbb{E}\left[(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_z^{-})(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_z^{-})^{\top}\right]$. The nonlinear update involves setting $\hat{\boldsymbol{x}}_z{}^{+} = \hat{\tilde{\mathbf{x}}}_z^{-} \oplus \hat{\boldsymbol{x}}_0{}^{-}$, $\hat{\tilde{\mathbf{x}}}_z^{+} = 0$, and $\mathbf{P}_z^{+} = \mathbf{P}_z^{-}$. Immediately following the propagation step, $\hat{\tilde{\mathbf{x}}}_0^{-}$ will still be $\mathbf{0}$. The Kalman update using measurement 1 is of the form:

$$\begin{aligned}
\hat{\tilde{\mathbf{x}}}_1^{-} &= \mathbf{K}_1\left(\boldsymbol{z}_1 \ominus \hat{\boldsymbol{z}}_1{}^{-}\right) \\
&\approx \mathbf{K}_1\left(\mathbf{H}_1\tilde{\mathbf{x}} + \mathbf{L}_1\mathbf{v}_1\right) \\
&\approx \hat{\tilde{\mathbf{x}}}_0^{-} + \mathbf{K}_1\left(\mathbf{H}_1\left(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_0^{-}\right) + \mathbf{L}_1\mathbf{v}_1\right)
\end{aligned}$$

The error in the error state estimates is given by:

$$\begin{aligned}
\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_1^{-} &\approx \tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_0^{-} - \mathbf{K}_1\left(\mathbf{H}_1\left(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_0^{-}\right) + \mathbf{L}_1\mathbf{v}_1\right) \\
&\approx \left(\mathbf{I} - \mathbf{K}_1\mathbf{H}_1\right)\left(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_0^{-}\right) - \mathbf{K}_1\mathbf{L}_1\mathbf{v}_1
\end{aligned}$$

And so in general the updated covariance $\mathbb{E}\left[\left(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_1^{-}\right)\left(\tilde{\mathbf{x}} - \hat{\tilde{\mathbf{x}}}_1^{-}\right)^{\top}\right] = \mathbf{P}_1^{-}$ is given by:

$$\mathbf{P}_1^{-} = \mathbf{P}_0^{-} - \mathbf{K}_1\mathbf{H}_1\mathbf{P}_0^{-} - \mathbf{P}_0^{-}\mathbf{H}_1^{\top}\mathbf{K}_1^{\top} + \mathbf{K}_1\mathbf{H}_1\mathbf{P}_0^{-}\mathbf{H}_1^{\top}\mathbf{K}_1^{\top} + \mathbf{K}_1\mathbf{L}_1\mathbf{L}_1^{\top}\mathbf{K}_1$$

If there is more than one uncorrelated measurement available at the current timestep, there are three options, and no obvious consensus in the literature as to which is the best. Option 1 would be to process it at the same time as measurement 1. That is, the measurement equation would be

$$\underbrace{\begin{bmatrix} \boldsymbol{z}_1 \ominus \hat{\boldsymbol{z}}_1 \\ \boldsymbol{z}_2 \ominus \hat{\boldsymbol{z}}_2 \end{bmatrix}}_{\tilde{\mathbf{z}}_{1,2}} \approx \underbrace{\begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{bmatrix}}_{\mathbf{H}_{1,2}} \left(\tilde{\mathbf{x}} - \hat{\hat{\mathbf{x}}}_0^-\right) + \underbrace{\begin{bmatrix} \mathbf{L}_1 \mathbf{v}_1 \\ \mathbf{L}_2 \mathbf{v}_2 \end{bmatrix}}_{\mathbf{L}_{1,2} \mathbf{v}_{1,2}},$$

and the update computes $\hat{\hat{\mathbf{x}}}_2^-$ and $\mathbf{P}_2^-$ as a function of $\tilde{\mathbf{z}}_{1,2}, \mathbf{H}_{1,2}, \mathbf{L}_{1,2}$ and $\mathbf{P}_0^-$. Using this option, the Jacobians are all computed around $\hat{\boldsymbol{x}}_0{}^-$.

Option 2 would be to process measurement 1 to obtain $\hat{\hat{\mathbf{x}}}_1^-, \mathbf{P}_1^-$, and then process the second measurement as

$$\begin{aligned} \hat{\hat{\mathbf{x}}}_2^- &= \hat{\hat{\mathbf{x}}}_1^- + \mathbf{K}_2 \left(\boldsymbol{z}_2 \ominus \hat{\boldsymbol{z}}_2{}^-\right) \\ &\approx \hat{\hat{\mathbf{x}}}_1^- + \mathbf{K}_2 \left(\mathbf{H}_2 \left(\tilde{\mathbf{x}} - \hat{\hat{\mathbf{x}}}_0^-\right) + \mathbf{L}_2 \mathbf{v}_2\right). \end{aligned}$$

In option 2, the Jacobians are still computed around $\hat{\boldsymbol{x}}_0{}^-$, but the Kalman gain is computed using $\mathbf{P}_1^-$. The measurement errors $\mathbf{L}_1\mathbf{v}_1$ and $\mathbf{L}_2\mathbf{v}_2$ must be uncorrelated for this to be valid. In the case where $\mathbf{K}_1$ and $\mathbf{K}_2$ are optimal, this produces an identical $\hat{\hat{\mathbf{x}}}_2^-$ and $\mathbf{P}_2^-$ as option 1,[1,2] which is easily seen in information form. It is unknown if this is still the case when $\mathbf{K}_1$ and $\mathbf{K}_2$ are suboptimal, as is the case with reduced order filters. The advantage of option 2 is that it uses less computational complexity to calculate the Kalman gains, since that involves inverting the residual covariance which has cubic complexity with the number of measurements being processed at once. This is the method used in the examples in Section 5.

In options 1 and 2, all of the measurements available at the given timestep are processed to produce a final $\hat{\hat{\mathbf{x}}}^-$ before it is used to perform the nonlinear correction to $\hat{\boldsymbol{x}}$. Option 3 is to perform the nonlinear update after each measurement is processed. Using this option the measurements would be linearized around the most up-to-date state estimates. The first measurement, $\boldsymbol{z}_1, \mathbf{H}_1, \mathbf{L}_1$ would still be functions of $\hat{\boldsymbol{x}}_0{}^-$. However, $\boldsymbol{z}_2, \mathbf{H}_2, \mathbf{L}_2$ would be functions of $\hat{\boldsymbol{x}}_1{}^+$, and in general $\boldsymbol{z}_z, \mathbf{H}_z, \mathbf{L}_z$ would be functions of $\hat{\boldsymbol{x}}_{z-1}{}^+$. The benefit of this option is that it *should* reduce the effect of linearization errors. The potential downside is that the effect of the linearization errors depends on which order the measurements are processed. Whether or not this is actually a problem is application specific.

3. System Partitioning

It is often advantageous to partition a system. During the propagation step, some error states evolve over time deterministically, while others do not. Error states that evolve over time such as position or orientation are referred to as *dynamic*. Error states such as biases or scale factor errors that do not change (except for random walks) are referred to as *static*. During the update step of a Kalman filter, some error states can be estimated while some can have their effects accounted for without explicitly estimating them. States that are updated in the innovation step of the Kalman filter are referred to as *active* states, while states that are not are referred to as *consider* states.

Partitioning the state vector in this way has two main advantages: stability and efficiency. By not multiplying the zero elements of the state transition matrix, the covariance propagation can be performed more efficiently in partitioned form if there are a large number of static error states. Likewise, consider states have their rows of the Kalman gain zeroed out to form a Schmidt Kalman filter,[1] which reduces overall complexity of the covariance update step if the number of consider states is large. In addition, using consider states allows the designer to completely model all of the known errors in a system, without effecting the observability of the system. Modeling all of the errors makes the system easier to tune, since neglecting these errors would require artificial inflation of the process and/or measurement covariance matrices. It also may provide some robustness to unknown or difficult-to-model errors. Using any variant of an EKF with an unobservable system is ill-advised because the first order Taylor series expansions become less accurate as the unobservable subspace accumulates error. The rest of this section will present the algebra involved in working with the partitioned system.

The four permutations of state types are referred to as $\tilde{\mathbf{a}}$, $\tilde{\mathbf{b}}$, $\tilde{\mathbf{c}}$, and $\tilde{\mathbf{d}}$ as defined in Fig. 1. The error-state vector and error covariance are partitioned as

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\mathbf{a}} \\ \tilde{\mathbf{b}} \\ \tilde{\mathbf{c}} \\ \tilde{\mathbf{d}} \end{bmatrix} \qquad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{aa} & \mathbf{P}_{ab} & \mathbf{P}_{ac} & \mathbf{P}_{ad} \\ \mathbf{P}_{ab}^\top & \mathbf{P}_{bb} & \mathbf{P}_{bc} & \mathbf{P}_{bd} \\ \mathbf{P}_{ac}^\top & \mathbf{P}_{bc}^\top & \mathbf{P}_{cc} & \mathbf{P}_{cd} \\ \mathbf{P}_{ad}^\top & \mathbf{P}_{bd}^\top & \mathbf{P}_{cd}^\top & \mathbf{P}_{dd} \end{bmatrix} \tag{10}$$

	dynamic	static
active	$\tilde{\mathrm{a}}$	$\tilde{\mathrm{b}}$
consider	$\tilde{\mathrm{c}}$	$\tilde{\mathrm{d}}$

Fig. 1 Error-state vector conventions

3.1 Discretization

The propagation equations can sometimes be carried out more efficiently using a discretized version of the partitioned system.

Assumptions:

- Dynamic consider state propagation does not depend on any other states
  - For example, correlated noise processes do not depend on states or biases
- The noise of a/b/c/d type processes is uncorrelated
  - For example, rate random walk is uncorrelated with angle random walk, etc.

This means the error-state derivatives in Eq. 6 will be partitioned as:

$$\dot{\tilde{\mathrm{x}}} = \begin{bmatrix} \dot{\tilde{\mathrm{a}}} \\ \dot{\tilde{\mathrm{b}}} \\ \dot{\tilde{\mathrm{c}}} \\ \dot{\tilde{\mathrm{d}}} \end{bmatrix} \approx \begin{bmatrix} \mathbf{A}_{aa} & \mathbf{A}_{ab} & \mathbf{A}_{ac} & \mathbf{A}_{ad} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathrm{a}} \\ \tilde{\mathrm{b}} \\ \tilde{\mathrm{c}} \\ \tilde{\mathrm{d}} \end{bmatrix} + \begin{bmatrix} \mathbf{G}_{aa} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{bb} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}_{dd} \end{bmatrix} \mathbf{w}$$

Carrying out the first few powers of $\mathbf{A}$ used in the matrix exponential series expansion reveals which terms of the state transition matrix will be non-zero:

$$\mathbf{A}^2 = \begin{bmatrix} \mathbf{A}_{aa}\mathbf{A}_{aa} & \mathbf{A}_{aa}\mathbf{A}_{ab} & \mathbf{A}_{aa}\mathbf{A}_{ac} + \mathbf{A}_{ac}\mathbf{A}_{cc} & \mathbf{A}_{aa}\mathbf{A}_{ad} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{cc}\mathbf{A}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{A}^3 = \begin{bmatrix} \mathbf{A}_{aa}\mathbf{A}_{aa}\mathbf{A}_{aa} & \mathbf{A}_{aa}\mathbf{A}_{aa}\mathbf{A}_{ab} & \mathbf{A}_{aa}\mathbf{A}_{aa}\mathbf{A}_{ac} + \big(\mathbf{A}_{aa}\mathbf{A}_{ac} + \mathbf{A}_{ac}\mathbf{A}_{cc}\big)\mathbf{A}_{cc} & \mathbf{A}_{aa}\mathbf{A}_{aa}\mathbf{A}_{ad} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{cc}\mathbf{A}_{cc}\mathbf{A}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

So, the state transition matrix is going to be of the form:

$$\mathbf{\Phi} = \begin{bmatrix} \mathbf{\Phi}_{aa} & \mathbf{\Phi}_{ab} & \mathbf{\Phi}_{ac} & \mathbf{\Phi}_{ad} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$$

What about the **Q** matrix? Evaluating the integral in Eq. 8 involves the integration of $\mathbf{\Phi}\mathbf{S}\mathbf{\Phi}^\top$, where $\mathbf{S} = \mathbf{G}\mathbf{G}^\top$. Because the partitioned **G** is block diagonal, so is **S**. The integrand takes the form:

$$\begin{aligned} \mathbf{\Phi}\mathbf{S}\mathbf{\Phi}^\top &= \begin{bmatrix} \mathbf{\Phi}_{aa} & \mathbf{\Phi}_{ab} & \mathbf{\Phi}_{ac} & \mathbf{\Phi}_{ad} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{S}_{aa} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{bb} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{dd} \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_{aa}^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_{ab}^\top & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_{ac}^\top & \mathbf{0} & \mathbf{\Phi}_{cc}^\top & \mathbf{0} \\ \mathbf{\Phi}_{ad}^\top & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{\Phi}_{aa}\mathbf{S}_{aa}\mathbf{\Phi}_{aa}^\top + \mathbf{\Phi}_{ab}\mathbf{S}_{bb}\mathbf{\Phi}_{ab}^\top + \mathbf{\Phi}_{ac}\mathbf{S}_{cc}\mathbf{\Phi}_{ac}^\top + \mathbf{\Phi}_{ad}\mathbf{S}_{dd}\mathbf{\Phi}_{ad}^\top & \mathbf{\Phi}_{ab}\mathbf{S}_{bb} & \mathbf{\Phi}_{ac}\mathbf{S}_{cc}\mathbf{\Phi}_{cc}^\top & \mathbf{\Phi}_{ad}\mathbf{S}_{dd} \\ \mathbf{S}_{bb}\mathbf{\Phi}_{ab}^\top & \mathbf{S}_{bb} & \mathbf{0} & \mathbf{0} \\ \mathbf{\Phi}_{cc}\mathbf{S}_{cc}\mathbf{\Phi}_{ac}^\top & \mathbf{0} & \mathbf{\Phi}_{cc}\mathbf{S}_{cc}\mathbf{\Phi}_{cc}^\top & \mathbf{0} \\ \mathbf{S}_{dd}\mathbf{\Phi}_{ad}^\top & \mathbf{0} & \mathbf{0} & \mathbf{S}_{dd} \end{bmatrix} \end{aligned}$$

The non-zero elements of the $\mathbf{Q}$ matrix will then be:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{aa} & \mathbf{Q}_{ab} & \mathbf{Q}_{ac} & \mathbf{Q}_{ad} \\ \mathbf{Q}_{ab}^{\top} & \mathbf{Q}_{bb} & \mathbf{0} & \mathbf{0} \\ \mathbf{Q}_{ac}^{\top} & \mathbf{0} & \mathbf{Q}_{cc} & \mathbf{0} \\ \mathbf{Q}_{ad}^{\top} & \mathbf{0} & \mathbf{0} & \mathbf{Q}_{dd} \end{bmatrix}$$

3.2 Propagation

The propagation of the covariance matrix from one timestep to the next is given by $\mathbf{P}^{-} = \mathbf{\Phi}\mathbf{P}\mathbf{\Phi}^{\top} + \mathbf{Q}$ (dropping subscripts). Expanding this in partitioned form leads to

$$\begin{aligned}
\mathbf{P}_{aa}{}^{-} &= \mathbf{\Phi}_{aa}\big(\mathbf{P}_{aa}\mathbf{\Phi}_{aa}^{\top} + \mathbf{P}_{ab}\mathbf{\Phi}_{ab}^{\top} + \mathbf{P}_{ac}\mathbf{\Phi}_{ac}^{\top} + \mathbf{P}_{ad}\mathbf{\Phi}_{ad}^{\top}\big) \\
&\quad + \mathbf{\Phi}_{ab}\big(\mathbf{P}_{ab}^{\top}\mathbf{\Phi}_{aa}^{\top} + \mathbf{P}_{bb}\mathbf{\Phi}_{ab}^{\top} + \mathbf{P}_{bc}\mathbf{\Phi}_{ac}^{\top} + \mathbf{P}_{bd}\mathbf{\Phi}_{ad}^{\top}\big) \\
&\quad + \mathbf{\Phi}_{ac}\big(\mathbf{P}_{ac}^{\top}\mathbf{\Phi}_{aa}^{\top} + \mathbf{P}_{bc}^{\top}\mathbf{\Phi}_{ab}^{\top} + \mathbf{P}_{cc}\mathbf{\Phi}_{ac}^{\top} + \mathbf{P}_{cd}\mathbf{\Phi}_{ad}^{\top}\big) \\
&\quad + \mathbf{\Phi}_{ad}\big(\mathbf{P}_{ad}^{\top}\mathbf{\Phi}_{aa}^{\top} + \mathbf{P}_{bd}^{\top}\mathbf{\Phi}_{ab}^{\top} + \mathbf{P}_{cd}^{\top}\mathbf{\Phi}_{ac}^{\top} + \mathbf{P}_{dd}\mathbf{\Phi}_{ad}^{\top}\big) \\
&\quad + \mathbf{Q}_{aa} \\
\mathbf{P}_{ab}{}^{-} &= \mathbf{\Phi}_{aa}\mathbf{P}_{ab} + \mathbf{\Phi}_{ab}\mathbf{P}_{bb} + \mathbf{\Phi}_{ac}\mathbf{P}_{bc}^{\top} + \mathbf{\Phi}_{ad}\mathbf{P}_{bd}^{\top} + \mathbf{Q}_{ab} \\
\mathbf{P}_{ac}{}^{-} &= \mathbf{\Phi}_{aa}\mathbf{P}_{ac}\mathbf{\Phi}_{cc}^{\top} + \mathbf{\Phi}_{ab}\mathbf{P}_{bc}\mathbf{\Phi}_{cc}^{\top} + \mathbf{\Phi}_{ac}\mathbf{P}_{cc}\mathbf{\Phi}_{cc}^{\top} + \mathbf{\Phi}_{ad}\mathbf{P}_{cd}^{\top}\mathbf{\Phi}_{cc}^{\top} + \mathbf{Q}_{ac} \\
\mathbf{P}_{ad}{}^{-} &= \mathbf{\Phi}_{aa}\mathbf{P}_{ad} + \mathbf{\Phi}_{ab}\mathbf{P}_{bd} + \mathbf{\Phi}_{ac}\mathbf{P}_{cd} + \mathbf{\Phi}_{ad}\mathbf{P}_{dd} + \mathbf{Q}_{ad} \\
\mathbf{P}_{bb}{}^{-} &= \mathbf{P}_{bb} + \mathbf{Q}_{bb} \\
\mathbf{P}_{bc}{}^{-} &= \mathbf{P}_{bc}\mathbf{\Phi}_{cc}^{\top} \\
\mathbf{P}_{bd}{}^{-} &= \mathbf{P}_{bd} \\
\mathbf{P}_{cc}{}^{-} &= \mathbf{\Phi}_{cc}\mathbf{P}_{cc}\mathbf{\Phi}_{cc}^{\top} + \mathbf{Q}_{cc} \\
\mathbf{P}_{cd}{}^{-} &= \mathbf{\Phi}_{cc}\mathbf{P}_{cd} \\
\mathbf{P}_{dd}{}^{-} &= \mathbf{P}_{dd} + \mathbf{Q}_{dd}.
\end{aligned}$$

3.3 Kalman Update

The next section will discuss combining multiple systems into a filter, but for now we will pretend that this has already occurred and we are working with a single filter with no external states. The residual error is partitioned as

$$\tilde{\mathbf{y}} \approx \begin{bmatrix}\mathbf{H}_a & \mathbf{H}_b & \mathbf{H}_c & \mathbf{H}_d\end{bmatrix} \begin{bmatrix}\tilde{\mathrm{a}} \\ \tilde{\mathrm{b}} \\ \tilde{\mathrm{c}} \\ \tilde{\mathrm{d}}\end{bmatrix} + \mathbf{L}\mathbf{v}.$$

The residual covariance matrix is given by:

$$\begin{aligned}\mathbf{S} = \mathbf{H}_a\big(&\mathbf{P}_{aa}\mathbf{H}_a^\top + \mathbf{P}_{ab}\mathbf{H}_b^\top + \mathbf{P}_{ac}\mathbf{H}_c^\top + \mathbf{P}_{ad}\mathbf{H}_d^\top\big)\\ &+ \mathbf{H}_b\big(\mathbf{P}_{ab}^\top\mathbf{H}_a^\top + \mathbf{P}_{bb}\mathbf{H}_b^\top + \mathbf{P}_{bc}\mathbf{H}_c^\top + \mathbf{P}_{bd}\mathbf{H}_d^\top\big)\\ &+ \mathbf{H}_c\big(\mathbf{P}_{ac}^\top\mathbf{H}_a^\top + \mathbf{P}_{bc}^\top\mathbf{H}_b^\top + \mathbf{P}_{cc}\mathbf{H}_c^\top + \mathbf{P}_{cd}\mathbf{H}_d^\top\big)\\ &+ \mathbf{H}_d\big(\mathbf{P}_{ad}^\top\mathbf{H}_a^\top + \mathbf{P}_{bd}^\top\mathbf{H}_b^\top + \mathbf{P}_{cd}^\top\mathbf{H}_c^\top + \mathbf{P}_{dd}\mathbf{H}_d^\top\big)\\ &+ \mathbf{L}\mathbf{L}^\top\end{aligned}$$

The Kalman gain is

$$\mathbf{K} = \mathbf{P}\mathbf{H}^\top\mathbf{S}^{-1},$$

which, when expanded gives

$$\begin{aligned}\mathbf{K}_a &= \big(\mathbf{P}_{aa}\mathbf{H}_a^\top + \mathbf{P}_{ab}\mathbf{H}_b^\top + \mathbf{P}_{ac}\mathbf{H}_c^\top + \mathbf{P}_{ad}\mathbf{H}_d^\top\big)\mathbf{S}^{-1}\\ \mathbf{K}_b &= \big(\mathbf{P}_{ab}^\top\mathbf{H}_a^\top + \mathbf{P}_{bb}\mathbf{H}_b^\top + \mathbf{P}_{bc}\mathbf{H}_c^\top + \mathbf{P}_{bd}\mathbf{H}_d^\top\big)\mathbf{S}^{-1}\\ \mathbf{K}_c &= \big(\mathbf{P}_{ac}^\top\mathbf{H}_a^\top + \mathbf{P}_{bc}^\top\mathbf{H}_b^\top + \mathbf{P}_{cc}\mathbf{H}_c^\top + \mathbf{P}_{cd}\mathbf{H}_d^\top\big)\mathbf{S}^{-1}\\ \mathbf{K}_d &= \big(\mathbf{P}_{ad}^\top\mathbf{H}_a^\top + \mathbf{P}_{bd}^\top\mathbf{H}_b^\top + \mathbf{P}_{cd}^\top\mathbf{H}_c^\top + \mathbf{P}_{dd}\mathbf{H}_d^\top\big)\mathbf{S}^{-1}.\end{aligned}$$

In the Schmidt Kalman filter construct, a suboptimal Kalman gain $\mathcal{K}$ structure is chosen in which the gain terms associated with the consider states are set to **0**. Using this suboptimal gain means we must use the Joseph form of the Kalman update

$$\begin{aligned}\mathbf{P}^+ &= (\mathbf{I} - \mathcal{K}\mathbf{H})\,\mathbf{P}\,(\mathbf{I} - \mathcal{K}\mathbf{H})^\top + \mathcal{K}\mathbf{R}\mathcal{K}^\top\\ &= \mathbf{P} - \mathcal{K}\mathbf{H}\mathbf{P} - \mathbf{P}\mathbf{H}^\top\mathcal{K}^\top + \mathcal{K}\mathbf{H}\mathbf{P}\mathbf{H}^\top\mathcal{K}^\top + \mathcal{K}\mathbf{R}\mathcal{K}^\top\\ &= \mathbf{P} - \mathcal{K}\mathbf{H}\mathbf{P} - \mathbf{P}\mathbf{H}^\top\mathcal{K}^\top + \mathcal{K}\mathbf{S}\mathcal{K}^\top.\end{aligned}$$

The Joseph form update can be simplified with a Schmidt Kalman gain, just not quite as much as for an optimal Kalman gain. The simplifications available are more apparent when using the follow substitutions:

$$\begin{aligned}
\mathcal{K} &= \begin{bmatrix} \mathcal{K}_\alpha \\ \mathbf{0} \end{bmatrix} & & \\
\mathcal{K}_\alpha &= \begin{bmatrix} \mathbf{K}_a \\ \mathbf{K}_b \end{bmatrix} & \mathbf{P}_{\alpha\alpha} &= \begin{bmatrix} \mathbf{P}_{aa} & \mathbf{P}_{ab} \\ \mathbf{P}_{ab}^\top & \mathbf{P}_{bb} \end{bmatrix} \\
\mathbf{P}_{\alpha\beta} &= \begin{bmatrix} \mathbf{P}_{ac} & \mathbf{P}_{ad} \\ \mathbf{P}_{bc} & \mathbf{P}_{bd} \end{bmatrix} & \mathbf{P}_{\beta\beta} &= \begin{bmatrix} \mathbf{P}_{cc} & \mathbf{P}_{cd} \\ \mathbf{P}_{cd}^\top & \mathbf{P}_{dd} \end{bmatrix} \\
\mathbf{H}_\alpha &= \begin{bmatrix} \mathbf{H}_a & \mathbf{H}_b \end{bmatrix} & \mathbf{H}_\beta &= \begin{bmatrix} \mathbf{H}_c & \mathbf{H}_d \end{bmatrix}.
\end{aligned}$$

The updated covariance is given by

$$\begin{aligned}
\mathbf{P}^+ = &\begin{bmatrix} \mathbf{P}_{\alpha\alpha} & \mathbf{P}_{\alpha\beta} \\ \mathbf{P}_{\alpha\beta}^\top & \mathbf{P}_{\beta\beta} \end{bmatrix} \\
&- \begin{bmatrix} \left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\left(\mathbf{H}_\alpha\mathbf{P}_{\alpha\alpha} + \mathbf{H}_\beta\mathbf{P}_{\alpha\beta}^\top\right) & \left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\left(\mathbf{H}_\alpha\mathbf{P}_{\alpha\beta} + \mathbf{H}_\beta\mathbf{P}_{\beta\beta}\right) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\
&- \begin{bmatrix} \left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)^\top & \mathbf{0} \\ \left(\mathbf{P}_{\alpha\beta}^\top\mathbf{H}_\alpha^\top + \mathbf{P}_{\beta\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)^\top & \mathbf{0} \end{bmatrix} \\
&+ \begin{bmatrix} \left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\mathbf{S}\mathbf{S}^{-1}\left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.
\end{aligned}$$

Noting the cancellation in the $\mathbf{P}_{\alpha\alpha}$ update, the partitioned covariance updates are given by

$$\begin{aligned}
{\mathbf{P}_{\alpha\alpha}}^+ &= \left(\mathbf{P}_{\alpha\alpha} - \left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\left(\mathbf{H}_\alpha\mathbf{P}_{\alpha\alpha} + \mathbf{H}_\beta\mathbf{P}_{\alpha\beta}^\top\right)\right) \\
{\mathbf{P}_{\alpha\beta}}^+ &= \left(\mathbf{P}_{\alpha\beta} - \left(\mathbf{P}_{\alpha\alpha}\mathbf{H}_\alpha^\top + \mathbf{P}_{\alpha\beta}\mathbf{H}_\beta^\top\right)\mathbf{S}^{-1}\left(\mathbf{H}_\alpha\mathbf{P}_{\alpha\beta} + \mathbf{H}_\beta\mathbf{P}_{\beta\beta}\right)\right) \\
{\mathbf{P}_{\beta\beta}}^+ &= \mathbf{P}_{\beta\beta}
\end{aligned}$$

or, in terms of $\mathcal{K}$

$$\begin{aligned}
\mathbf{P}_{\alpha\alpha}^+ &= \mathbf{P}_{\alpha\alpha} - \mathcal{K}_\alpha\left(\mathbf{H}_\alpha\mathbf{P}_{\alpha\alpha} + \mathbf{H}_\beta\mathbf{P}_{\alpha\beta}^\top\right) \\
\mathbf{P}_{\alpha\beta}^+ &= \mathbf{P}_{\alpha\beta} - \mathcal{K}_\alpha\left(\mathbf{H}_\alpha\mathbf{P}_{\alpha\beta} + \mathbf{H}_\beta\mathbf{P}_{\beta\beta}\right) \\
\mathbf{P}_{\beta\beta}^+ &= \mathbf{P}_{\beta\beta}.
\end{aligned}$$

Expanding the updates leads to

$$
\begin{aligned}
\mathbf{P}_{aa}{}^{+} &= \mathbf{P}_{aa} - \mathbf{K}_a\big(\mathbf{H}_a\mathbf{P}_{aa} + \mathbf{H}_b\mathbf{P}_{ab}^{\top}\big) - \mathbf{K}_a\big(\mathbf{H}_c\mathbf{P}_{ac}^{\top} + \mathbf{H}_d\mathbf{P}_{ad}^{\top}\big) \\
\mathbf{P}_{ab}{}^{+} &= \mathbf{P}_{ab} - \mathbf{K}_a\big(\mathbf{H}_a\mathbf{P}_{ab} + \mathbf{H}_b\mathbf{P}_{bb}\big) - \mathbf{K}_a\big(\mathbf{H}_c\mathbf{P}_{bc}^{\top} + \mathbf{H}_d\mathbf{P}_{bd}^{\top}\big) \\
\mathbf{P}_{bb}{}^{+} &= \mathbf{P}_{bb} - \mathbf{K}_b\big(\mathbf{H}_a\mathbf{P}_{ab} + \mathbf{H}_b\mathbf{P}_{bb}\big) - \mathbf{K}_b\big(\mathbf{H}_c\mathbf{P}_{bc}^{\top} + \mathbf{H}_d\mathbf{P}_{bd}^{\top}\big) \\
\mathbf{P}_{ac}{}^{+} &= \mathbf{P}_{ac} - \mathbf{K}_a\big(\mathbf{H}_a\mathbf{P}_{ac} + \mathbf{H}_b\mathbf{P}_{bc}\big) - \mathbf{K}_a\big(\mathbf{H}_c\mathbf{P}_{cc} + \mathbf{H}_d\mathbf{P}_{cd}^{\top}\big) \\
\mathbf{P}_{ad}{}^{+} &= \mathbf{P}_{ad} - \mathbf{K}_a\big(\mathbf{H}_a\mathbf{P}_{ad} + \mathbf{H}_b\mathbf{P}_{bd}\big) - \mathbf{K}_a\big(\mathbf{H}_c\mathbf{P}_{cd} + \mathbf{H}_d\mathbf{P}_{dd}\big) \\
\mathbf{P}_{bc}{}^{+} &= \mathbf{P}_{bc} - \mathbf{K}_b\big(\mathbf{H}_a\mathbf{P}_{ac} + \mathbf{H}_b\mathbf{P}_{bc}\big) - \mathbf{K}_b\big(\mathbf{H}_c\mathbf{P}_{cc} + \mathbf{H}_d\mathbf{P}_{cd}^{\top}\big) \\
\mathbf{P}_{bd}{}^{+} &= \mathbf{P}_{bd} - \mathbf{K}_b\big(\mathbf{H}_a\mathbf{P}_{ad} + \mathbf{H}_b\mathbf{P}_{bd}\big) - \mathbf{K}_b\big(\mathbf{H}_c\mathbf{P}_{cd} + \mathbf{H}_d\mathbf{P}_{dd}\big).
\end{aligned}
$$

3.3.1 Reduced Computation

The Kalman gain and covariance updates can be written in a slightly less computationally complex form using partitioned matrices **U** and **V** as intermediate values:

$$
\begin{aligned}
\mathbf{U}_a &= \mathbf{P}_{aa}\mathbf{H}_a^{\top} + \mathbf{P}_{ab}\mathbf{H}_b^{\top} + \mathbf{P}_{ac}\mathbf{H}_c^{\top} + \mathbf{P}_{ad}\mathbf{H}_d^{\top} \\
\mathbf{U}_b &= \mathbf{P}_{ab}^{\top}\mathbf{H}_a^{\top} + \mathbf{P}_{bb}\mathbf{H}_b^{\top} + \mathbf{P}_{bc}\mathbf{H}_c^{\top} + \mathbf{P}_{bd}\mathbf{H}_d^{\top} \\
\mathbf{U}_c &= \mathbf{P}_{ac}^{\top}\mathbf{H}_a^{\top} + \mathbf{P}_{bc}^{\top}\mathbf{H}_b^{\top} + \mathbf{P}_{cc}\mathbf{H}_c^{\top} + \mathbf{P}_{cd}\mathbf{H}_d^{\top} \\
\mathbf{U}_d &= \mathbf{P}_{ad}^{\top}\mathbf{H}_a^{\top} + \mathbf{P}_{bd}^{\top}\mathbf{H}_b^{\top} + \mathbf{P}_{cd}^{\top}\mathbf{H}_c^{\top} + \mathbf{P}_{dd}\mathbf{H}_d^{\top}
\end{aligned}
$$

$$
\mathbf{S} = \mathbf{H}_a\mathbf{U}_a + \mathbf{H}_b\mathbf{U}_b + \mathbf{H}_c\mathbf{U}_c + \mathbf{H}_d\mathbf{U}_d + \mathbf{L}\mathbf{L}^{\top}
$$

$$
\begin{aligned}
\mathbf{K}_a &= \mathbf{U}_a\mathbf{S}^{-1} \\
\mathbf{K}_b &= \mathbf{U}_b\mathbf{S}^{-1}
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{V}_{aa} &= \mathbf{K}_a\mathbf{U}_a^\top \\
\mathbf{V}_{ab} &= \mathbf{K}_a\mathbf{U}_b^\top \\
\mathbf{V}_{ac} &= \mathbf{K}_a\mathbf{U}_c^\top \\
\mathbf{V}_{ad} &= \mathbf{K}_a\mathbf{U}_d^\top \\
\mathbf{V}_{ba} &= \mathbf{K}_b\mathbf{U}_a^\top \\
\mathbf{V}_{bb} &= \mathbf{K}_b\mathbf{U}_b^\top \\
\mathbf{V}_{bc} &= \mathbf{K}_b\mathbf{U}_c^\top \\
\mathbf{V}_{bd} &= \mathbf{K}_b\mathbf{U}_d^\top
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{P}_{aa}{}^+ &= \mathbf{P}_{aa} - \mathbf{V}_{aa} \\
\mathbf{P}_{ab}{}^+ &= \mathbf{P}_{ab} - \mathbf{V}_{ab} \\
\mathbf{P}_{ac}{}^+ &= \mathbf{P}_{ac} - \mathbf{V}_{ac} \\
\mathbf{P}_{ad}{}^+ &= \mathbf{P}_{ad} - \mathbf{V}_{ad} \\
\mathbf{P}_{bb}{}^+ &= \mathbf{P}_{bb} - \mathbf{V}_{bb} \\
\mathbf{P}_{bc}{}^+ &= \mathbf{P}_{bc} - \mathbf{V}_{bc} \\
\mathbf{P}_{bd}{}^+ &= \mathbf{P}_{bd} - \mathbf{V}_{bd} \\
\mathbf{P}_{cc}{}^+ &= \mathbf{P}_{cc} \\
\mathbf{P}_{cd}{}^+ &= \mathbf{P}_{cd} \\
\mathbf{P}_{dd}{}^+ &= \mathbf{P}_{dd}
\end{aligned}
$$

3.3.2 Joseph Form

In certain situations, such as when computing an error budget, it is necessary to update a filter using a completely arbitrary gain matrix. In these cases, the Joseph form still gives the updated covariance without requiring any assumptions on how the gain was calculated.[11] Using new partitioned matrices **U**, **V**, **W**, and **X** as intermediate values, the update becomes:

$$
\begin{aligned}
\mathbf{U}_a &= \mathbf{H}_a\mathbf{P}_{aa} + \mathbf{H}_b\mathbf{P}_{ab}^\top + \mathbf{H}_c\mathbf{P}_{ac}^\top + \mathbf{H}_d\mathbf{P}_{ad}^\top \\
\mathbf{U}_b &= \mathbf{H}_a\mathbf{P}_{ab} + \mathbf{H}_b\mathbf{P}_{bb} + \mathbf{H}_c\mathbf{P}_{bc}^\top + \mathbf{H}_d\mathbf{P}_{bd}^\top \\
\mathbf{U}_c &= \mathbf{H}_a\mathbf{P}_{ac} + \mathbf{H}_b\mathbf{P}_{bc} + \mathbf{H}_c\mathbf{P}_{cc} + \mathbf{H}_d\mathbf{P}_{cd}^\top \\
\mathbf{U}_d &= \mathbf{H}_a\mathbf{P}_{ad} + \mathbf{H}_b\mathbf{P}_{bd} + \mathbf{H}_c\mathbf{P}_{cd} + \mathbf{H}_d\mathbf{P}_{dd}
\end{aligned}
$$

$$\begin{aligned}
\mathbf{V}_{aa} &= \mathbf{K}_a\mathbf{U}_a \\
\mathbf{V}_{ab} &= \mathbf{K}_a\mathbf{U}_b \\
\mathbf{V}_{ac} &= \mathbf{K}_a\mathbf{U}_c \\
\mathbf{V}_{ad} &= \mathbf{K}_a\mathbf{U}_d \\
\mathbf{V}_{ba} &= \mathbf{K}_b\mathbf{U}_a \\
\mathbf{V}_{bb} &= \mathbf{K}_b\mathbf{U}_b \\
\mathbf{V}_{bc} &= \mathbf{K}_b\mathbf{U}_c \\
\mathbf{V}_{bd} &= \mathbf{K}_b\mathbf{U}_d
\end{aligned}$$

$$\begin{aligned}
\mathbf{W}_a &= \mathbf{V}_{aa}\mathbf{H}_a^\top + \mathbf{V}_{ab}\mathbf{H}_b^\top + \mathbf{V}_{ac}\mathbf{H}_c^\top + \mathbf{V}_{ad}\mathbf{H}_d^\top \\
\mathbf{W}_b &= \mathbf{V}_{ba}\mathbf{H}_a^\top + \mathbf{V}_{bb}\mathbf{H}_b^\top + \mathbf{V}_{bc}\mathbf{H}_c^\top + \mathbf{V}_{bd}\mathbf{H}_d^\top
\end{aligned}$$

$$\begin{aligned}
\mathbf{X}_{aa} &= \mathbf{W}_a\mathbf{K}_a^\top \\
\mathbf{X}_{ab} &= \mathbf{W}_a\mathbf{K}_b^\top \\
\mathbf{X}_{bb} &= \mathbf{W}_b\mathbf{K}_b^\top
\end{aligned}$$

$$\begin{aligned}
{\mathbf{P}_{aa}}^+ &= \mathbf{P}_{aa} + \mathbf{X}_{aa} + \mathbf{K}_a\mathbf{L}\mathbf{L}^\top\mathbf{K}_a^\top - \mathbf{V}_{aa} - \mathbf{V}_{aa}^\top \\
{\mathbf{P}_{ab}}^+ &= \mathbf{P}_{ab} + \mathbf{X}_{ab} + \mathbf{K}_a\mathbf{L}\mathbf{L}^\top\mathbf{K}_b^\top - \mathbf{V}_{ab} - \mathbf{V}_{ba}^\top \\
{\mathbf{P}_{ac}}^+ &= \mathbf{P}_{ac} + \mathbf{X}_{ac} - \mathbf{V}_{ac} \\
{\mathbf{P}_{ad}}^+ &= \mathbf{P}_{ad} + \mathbf{X}_{ad} - \mathbf{V}_{ad} \\
{\mathbf{P}_{bb}}^+ &= \mathbf{P}_{bb} + \mathbf{X}_{bb} + \mathbf{K}_b\mathbf{L}\mathbf{L}^\top\mathbf{K}_b^\top - \mathbf{V}_{bb} - \mathbf{V}_{bb}^\top \\
{\mathbf{P}_{bc}}^+ &= \mathbf{P}_{bc} + \mathbf{X}_{bc} - \mathbf{V}_{bc} \\
{\mathbf{P}_{bd}}^+ &= \mathbf{P}_{bd} + \mathbf{X}_{bd} - \mathbf{V}_{bd} \\
{\mathbf{P}_{cc}}^+ &= \mathbf{P}_{cc} + \mathbf{X}_{cc} \\
{\mathbf{P}_{cd}}^+ &= \mathbf{P}_{cd} + \mathbf{X}_{cd} \\
{\mathbf{P}_{dd}}^+ &= \mathbf{P}_{dd} + \mathbf{X}_{dd}
\end{aligned}$$

3.4 Partitioned Accumulation

Some systems have higher sample rates than others. In some cases, it is helpful to "accumulate" state transition matrices and Q matrices to show equivalent transformations between updates. For example, in visual-inertial odometry problems it is intuitive to accumulate state transition matrices for states related to the IMU, and only update the error covariance when a camera frame is available.[12] Because IMUs are typically sampled at least 10 times faster than cameras, this leads to significant computational savings.

Consider two steps of covariance propagation:

$$\begin{aligned}
\mathbf{P}_{k+1} &= \mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^\top + \mathbf{Q}_k \\
\mathbf{P}_{k+2} &= \mathbf{\Phi}_{k+1} \mathbf{P}_{k+1} \mathbf{\Phi}_{k+1}^\top + \mathbf{Q}_{k+1} \\
&= \mathbf{\Phi}_{k+1} \big(\mathbf{\Phi}_k \mathbf{P}_k \mathbf{\Phi}_k^\top + \mathbf{Q}_k \big) \mathbf{\Phi}_{k+1}^\top + \mathbf{Q}_{k+1} \\
&= \big(\mathbf{\Phi}_{k+1} \mathbf{\Phi}_k \big) \mathbf{P}_k \big(\mathbf{\Phi}_{k+1} \mathbf{\Phi}_k \big)^\top + \mathbf{\Phi}_{k+1} \mathbf{Q}_k \mathbf{\Phi}_{k+1}^\top + \mathbf{Q}_{k+1}.
\end{aligned}$$

If multiple propagation steps occur in between measurement updates, an equivalent transformation from the last measurement time n up to the current timestep can be propagated with

$$\begin{aligned}
\mathbf{\Phi}_n^+ &= \mathbf{\Phi}_k \mathbf{\Phi}_n \\
\mathbf{Q}_n^+ &= \mathbf{\Phi}_k \mathbf{Q}_n \mathbf{\Phi}_k^\top + \mathbf{Q}_k.
\end{aligned}$$

Expanding these calculations for the partitioned form leads to

$$\begin{aligned}
\mathbf{\Phi}_{aa_n}^{\;+} &= \mathbf{\Phi}_{aa_k} \mathbf{\Phi}_{aa_n} \\
\mathbf{\Phi}_{ab_n}^{\;+} &= \mathbf{\Phi}_{aa_k} \mathbf{\Phi}_{ab_n} + \mathbf{\Phi}_{ab_k} \\
\mathbf{\Phi}_{ac_n}^{\;+} &= \mathbf{\Phi}_{aa_k} \mathbf{\Phi}_{ac_n} + \mathbf{\Phi}_{ac_k} \mathbf{\Phi}_{cc_n} \\
\mathbf{\Phi}_{ad_n}^{\;+} &= \mathbf{\Phi}_{aa_k} \mathbf{\Phi}_{ad_n} + \mathbf{\Phi}_{ad_k} \\
\mathbf{\Phi}_{cc_n}^{\;+} &= \mathbf{\Phi}_{cc_k} \mathbf{\Phi}_{cc_n},
\end{aligned}$$

and

$$\begin{aligned}
{\mathbf{Q}_{aa_n}}^+ &= \mathbf{\Phi}_{aa_k}\left(\mathbf{Q}_{aa_n}\mathbf{\Phi}_{aa_k}^\top + \mathbf{Q}_{ab_n}\mathbf{\Phi}_{ab_k}^\top + \mathbf{Q}_{ac_n}\mathbf{\Phi}_{ac_k}^\top + \mathbf{Q}_{ad_n}\mathbf{\Phi}_{ad_k}^\top\right)\\
&\quad + \mathbf{\Phi}_{ab_k}\left(\mathbf{Q}_{ab_n}^\top\mathbf{\Phi}_{aa_k}^\top + \mathbf{Q}_{bb_n}\mathbf{\Phi}_{ab_k}^\top\right)\\
&\quad + \mathbf{\Phi}_{ac_k}\left(\mathbf{Q}_{ac_n}^\top\mathbf{\Phi}_{aa_k}^\top + \mathbf{Q}_{cc_n}\mathbf{\Phi}_{ac_k}^\top\right)\\
&\quad + \mathbf{\Phi}_{ad_k}\left(\mathbf{Q}_{ad_n}^\top\mathbf{\Phi}_{aa_k}^\top + \mathbf{Q}_{dd_n}\mathbf{\Phi}_{ad_k}^\top\right)\\
&\quad + \mathbf{Q}_{aa_k}\\
{\mathbf{Q}_{ab_n}}^+ &= \mathbf{\Phi}_{aa_k}\mathbf{Q}_{ab_n} + \mathbf{\Phi}_{ab_k}\mathbf{Q}_{bb_n} + \mathbf{Q}_{ab_k}\\
{\mathbf{Q}_{ac_n}}^+ &= \mathbf{\Phi}_{aa_k}\mathbf{Q}_{ac_n}\mathbf{\Phi}_{cc_k}^\top + \mathbf{\Phi}_{ac_k}\mathbf{Q}_{cc_n}\mathbf{\Phi}_{cc_k}^\top + \mathbf{Q}_{ac_k}\\
{\mathbf{Q}_{ad_n}}^+ &= \mathbf{\Phi}_{aa_k}\mathbf{Q}_{ad_n} + \mathbf{\Phi}_{ad_k}\mathbf{Q}_{dd_n} + \mathbf{Q}_{ad_k}\\
{\mathbf{Q}_{bb_n}}^+ &= \mathbf{Q}_{bb_n} + \mathbf{Q}_{bb_k}\\
{\mathbf{Q}_{cc_n}}^+ &= \mathbf{\Phi}_{cc_k}\mathbf{Q}_{cc_n}\mathbf{\Phi}_{cc_k}^\top + \mathbf{Q}_{cc_k}\\
{\mathbf{Q}_{dd_n}}^+ &= \mathbf{Q}_{dd_n} + \mathbf{Q}_{dd_k}.
\end{aligned}$$

3.5 Stochastic Cloning

Stochastic cloning is a state transformation that is frequently used to process delayed measurements.[13] Typically the cloned states are stored as active, constant states, and are transformed from the active states only. Therefore, the linear transformation of the state errors is given by:

$$\begin{bmatrix} \tilde{\mathrm{a}}' \\ \tilde{\mathrm{b}}' \\ \tilde{\mathrm{c}}' \\ \tilde{\mathrm{d}}' \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{ba} & \mathbf{C}_{bb} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathrm{a}} \\ \tilde{\mathrm{b}} \\ \tilde{\mathrm{c}} \\ \tilde{\mathrm{d}} \end{bmatrix}$$

As such, the covariance matrix must be transformed:

$$
\begin{aligned}
\mathbf{P}'_{aa} &= \mathbf{P}_{aa} \\
\mathbf{P}'_{ab} &= \left(\mathbf{P}_{aa}\mathbf{C}_{ba}^{\top} + \mathbf{P}_{ab}\mathbf{C}_{bb}^{\top}\right) \\
\mathbf{P}'_{ac} &= \mathbf{P}_{ac} \\
\mathbf{P}'_{ad} &= \mathbf{P}_{ad} \\
\mathbf{P}'_{bb} &= \mathbf{C}_{ba}\left(\mathbf{P}_{aa}\mathbf{C}_{ba}^{\top} + \mathbf{P}_{ab}\mathbf{C}_{bb}^{\top}\right) \\
&\quad + \mathbf{C}_{bb}\left(\mathbf{P}_{ab}^{\top}\mathbf{C}_{ba}^{\top} + \mathbf{P}_{bb}\mathbf{C}_{bb}^{\top}\right) \\
\mathbf{P}'_{bc} &= \mathbf{C}_{ba}\mathbf{P}_{ac} + \mathbf{C}_{bb}\mathbf{P}_{bc} \\
\mathbf{P}'_{bd} &= \mathbf{C}_{ba}\mathbf{P}_{ad} + \mathbf{C}_{bb}\mathbf{P}_{bd} \\
\mathbf{P}'_{cc} &= \mathbf{P}_{cc} \\
\mathbf{P}'_{cd} &= \mathbf{P}_{cd} \\
\mathbf{P}'_{dd} &= \mathbf{P}_{dd}
\end{aligned}
$$

3.6 General a/b Transform

The following transform can be useful for when things are transformed from a states to b states.

$$
\begin{bmatrix} \mathrm{a}' \\ \mathrm{b}' \\ \mathrm{c}' \\ \mathrm{d}' \end{bmatrix} = \begin{bmatrix} \mathbf{G}_{aa} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{G}_{ba} & \mathbf{G}_{bb} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathrm{a} \\ \mathrm{b} \\ \mathrm{c} \\ \mathrm{d} \end{bmatrix}
$$

$$
\begin{aligned}
\mathbf{P}'_{aa} &= \mathbf{G}_{aa}\mathbf{P}_{aa}\mathbf{G}_{aa}^\top && (11)\\
\mathbf{P}'_{ab} &= \mathbf{G}_{aa}\big(\mathbf{P}_{aa}\mathbf{G}_{ba}^\top + \mathbf{P}_{ab}\mathbf{G}_{bb}^\top\big) && (12)\\
\mathbf{P}'_{ac} &= \mathbf{G}_{aa}\mathbf{P}_{ac} && (13)\\
\mathbf{P}'_{ad} &= \mathbf{G}_{aa}\mathbf{P}_{ad} && (14)\\
\mathbf{P}'_{bb} &= \big(\mathbf{G}_{ba}\big(\mathbf{P}_{aa}\mathbf{G}_{ba}^\top + \mathbf{P}_{ab}\mathbf{G}_{bb}^\top\big) && (15)\\
&\quad + \mathbf{G}_{bb}\big(\mathbf{P}_{ab}^\top\mathbf{G}_{ba}^\top + \mathbf{P}_{bb}\mathbf{G}_{bb}^\top\big)\big) && (16)\\
\mathbf{P}'_{bc} &= \big(\mathbf{G}_{ba}\mathbf{P}_{ac} + \mathbf{G}_{bb}\mathbf{P}_{bc}\big) && (17)\\
\mathbf{P}'_{bd} &= \big(\mathbf{G}_{ba}\mathbf{P}_{ad} + \mathbf{G}_{bb}\mathbf{P}_{bd}\big) && (18)\\
\mathbf{P}'_{cc} &= \mathbf{P}_{cc} && (19)\\
\mathbf{P}'_{cd} &= \mathbf{P}_{cd} && (20)\\
\mathbf{P}'_{dd} &= \mathbf{P}_{dd} && (21)
\end{aligned}
$$

3.7 Complexity

The complexity of most common operations, covariance propagation and covariance update, can be approximated by counting the number of multiplications. If a is the dimension of the active-dynamic partition of the whole filter (i.e., $\dim(\tilde{\mathbf{a}}_f)$), and b, c, d, the dimensions of their prospective partitions, then the sum of multiplications in the covariance propagation is given by:

$$
\begin{aligned}
\text{propagation complexity} = \mathcal{O}(&ab^2 + 3a^2b + 3ac^2 + 4a^2c + 2bc^2 + 3a^2d + 2c^2d\\
&+ 2a^3 + 2c^3 + 3abc + 2abd + 3acd + add')
\end{aligned}
$$

This accounting uses the fact that some of the terms that appear in the $\mathbf{P}_{aa}^-$ update are functions of $\mathbf{P}_{ab}^-$ and $\mathbf{P}_{ad}^-$, which could be computed first and then used in the computation of $\mathbf{P}_{aa}^-$. The term d' could be 1 or d, depending on the modeling of $\mathbf{P}_{dd}$. In many cases, $\mathbf{P}_{dd}$ will be initialized as a diagonal matrix, and it will stay that way through propagation and update operations. This sparsity can be exploited to make $\mathbf{P}_{dd}\mathbf{\Phi}_{ad}^\top$ a $\mathcal{O}(ad)$ operation instead of a $\mathcal{O}(ad^2)$ one.

Counting all of the multiplications in Section 3.3.1 involved in the Kalman update with a measurement vector of dimension m leads to:

$$\begin{aligned}\text{update complexity} = \mathcal{O}(m(&4ab + 3ac + 3ad + 3bc + 3bd + 2cd + dd' \\ &+ 2am + 2bm + cm + dm + 2a^2 + 2b^2 + c^2 + m^2))\end{aligned}$$

There is a m^3 term that refers to the $\mathcal{O}(m^3)$ cost of inverting **S** when computing the Kalman gain, even though this is not strictly multiplication.

It is possible that more simplifications exist that could reduce the computational burden. The naive Kalman filter with no partitioning uses $2n^3$ multiplications to perform covariance propagation, where $n = a + b + c + d$. The naive covariance update equation is actually more streamlined because the Joseph form update simplifies to $\mathbf{P}^+ = \mathbf{P} - \mathbf{KHP}$ when **K** is optimal. However, the Kalman gain is still a $\mathcal{O}(n^2m + 2nm^2 + m^3)$ operation and the covariance update another $\mathcal{O}(n^2m)$ operation.

4. Modular Kalman Filters

The point of a Kalman filter is to combine information from multiple systems. However, it is often helpful to develop systems independently so that they can be unit tested and re-used. Creating a filter consists of concatenating the partitioned state vectors of the subsystems to form an augmented system. That way, the partitioned operations (covariance propagation, Kalman update, etc.) described in the previous section can be performed on the augmented system.

The filter system f consists of an error state vector, state transition matrix, discrete Q matrix, and measurement Jacobians that are constructed from the individual systems; system 1, 2, 3, etc. The filter error-state vector consists of

$$\tilde{\mathbf{x}}_{a_f} = \begin{bmatrix}\tilde{\mathbf{x}}_{a_1} \\ \tilde{\mathbf{x}}_{a_2} \\ \vdots\end{bmatrix} \qquad \tilde{\mathbf{x}}_{b_f} = \begin{bmatrix}\tilde{\mathbf{x}}_{b_1} \\ \tilde{\mathbf{x}}_{b_2} \\ \vdots\end{bmatrix} \qquad \tilde{\mathbf{x}}_{c_f} = \begin{bmatrix}\tilde{\mathbf{x}}_{c_1} \\ \tilde{\mathbf{x}}_{c_2} \\ \vdots\end{bmatrix} \qquad \tilde{\mathbf{x}}_{d_f} = \begin{bmatrix}\tilde{\mathbf{x}}_{d_1} \\ \tilde{\mathbf{x}}_{d_2} \\ \vdots\end{bmatrix}.$$

The filter propagation matrices are block diagonal constructions of the component system propagation matrices:

$$\mathbf{\Phi}_{aa_f} = \begin{bmatrix} \mathbf{\Phi}_{aa_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{aa_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots \end{bmatrix} \qquad \mathbf{\Phi}_{ab_f} = \begin{bmatrix} \mathbf{\Phi}_{ab_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Phi}_{ab_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots \end{bmatrix} \qquad etc. \tag{22}$$

The external error-states that contribute to system outputs in Eq. 9 are now error-states from other systems in the filter. For example, a simple magnetometer system (mag) has an output that is dependent upon attitude errors. However, the only error-states in the simple magnetometer system are the bias errors, which are modeled as static active (b) states. The magnetometer system models the output residual as:

$$\tilde{\mathbf{z}}_{mag} \approx \mathbf{H}_{b_{mag}} \tilde{\mathbf{x}}_{b_{mag}} + \mathbf{E}_\theta \tilde{\mathbf{e}}_\theta + \mathbf{L}_{mag} \mathbf{v}_{mag},$$

where $\tilde{\mathbf{e}}_\theta$ is the external attitude errors. Suppose a filter f is constructed that combines an IMU system (IMU) with the simple magnetometer system. The attitude errors are part of the IMU system, which models attitude errors as dynamic active (a) states. The external attitude errors are now just $\tilde{\mathbf{e}}_\theta = \lceil \tilde{\mathbf{x}}_{a_{IMU}} \rceil_{\tilde{\boldsymbol{\theta}}} \tilde{\mathbf{x}}_{a_{IMU}}$, where $\lceil \tilde{\mathbf{x}}_{a_{IMU}} \rceil_{\tilde{\boldsymbol{\theta}}}$ is the Jacobian of the IMU error-state vector with respect to the attitude errors. For the Kalman filter update, the magnetometer output error becomes

$$\begin{aligned} \tilde{\mathbf{z}}_{mag} &\approx \mathbf{H}_{b_{mag}} \tilde{\mathbf{x}}_{b_{mag}} + \mathbf{E}_\theta \lceil \tilde{\mathbf{x}}_{a_{IMU}} \rceil_{\tilde{\boldsymbol{\theta}}} \tilde{\mathbf{x}}_{a_{IMU}} + \mathbf{L}_{mag} \mathbf{v}_{mag} \\ &\approx \underbrace{\begin{bmatrix} \mathbf{E}_\theta \lceil \tilde{\mathbf{x}}_{a_{IMU}} \rceil_{\tilde{\boldsymbol{\theta}}} & \mathbf{0} \end{bmatrix}}_{\mathbf{H}_{a_f}} \underbrace{\begin{bmatrix} \tilde{\mathbf{x}}_{a_{IMU}} \\ \tilde{\mathbf{x}}_{a_{mag}} \end{bmatrix}}_{\tilde{\mathbf{x}}_{a_f}} + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{H}_{b_{mag}} \end{bmatrix}}_{\mathbf{H}_{b_f}} \underbrace{\begin{bmatrix} \tilde{\mathbf{x}}_{b_{IMU}} \\ \tilde{\mathbf{x}}_{b_{mag}} \end{bmatrix}}_{\tilde{\mathbf{x}}_{b_f}} + \mathbf{L}_{mag} \mathbf{v}_{mag} \end{aligned}$$

5. Example

In this example, three example filters are constructed from three example systems. The majority of the states are present in an IMU system, but magnetometer systems and GPS system are incorporated to use their measurements. The goal is to describe the modeling of the systems as described in Section 2, then partition the systems in different ways. For each system, there will be an *oversimplified*, *optimal*, and *balanced* approach to this partitioning. Filters are created from the different partitioned systems and evaluated on reference trajectories.

5.1 Inertial Measurement Unit (IMU)

IMUs consist of an orthogonal triad of gyroscopes and accelerometers. They are the backbone of most navigation systems because they do not require the use of a physics-based dynamics model to propagate position, velocity, and attitude estimates.

5.1.1 System Modeling

Though it is possible to (approximately) separate the inertial sensor errors from the resulting trajectory errors (see Forster et al.[14]), it is simpler to include them in one dynamic system (see Titterton and Weston[10]).

The kinematics that drive the system assume a fixed, nonrotating global reference frame G, and a body-frame B that is centered at and rotates with the IMU. The states modeled by the IMU system are listed in Table 2. The external inputs to the system are the raw gyroscope and accelerometer outputs ${}^{B}\breve{\boldsymbol{\omega}}_{B/G}$ and ${}^{B}\breve{\mathbf{a}}$, and gravity ${}^{G}\mathbf{g}$, which is assumed to be known perfectly in global coordinates. The velocity and rotation errors depend on the gyroscope and accelerometer errors ${}^{B}\tilde{\boldsymbol{\omega}}_{B/G}$ and ${}^{B}\tilde{\mathbf{a}}$. In this case, ${}^{B}\breve{\boldsymbol{\omega}}_{B/G}$ and ${}^{B}\breve{\mathbf{a}}$ are modeled as functions of the true angular rate and specific force (${}^{B}\boldsymbol{\omega}_{B/G}$ and ${}^{B}\mathbf{a}$) modified by several factors:

$$\begin{bmatrix} {}^{B}\breve{\boldsymbol{\omega}}_{B/G} \\ {}^{B}\breve{\mathbf{a}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{I} + \lfloor \mathbf{s}_\omega \backslash \rfloor + \lfloor \mathbf{m}_\omega \times \rfloor + \lfloor \mathbf{n}_\omega \triangleleft \rfloor & \lfloor \mathbf{g}_\omega \equiv \rfloor \\ \mathbf{0} & \mathbf{I} + \lfloor \mathbf{s}_a \backslash \rfloor + \lfloor \mathbf{m}_a \times \rfloor + \lfloor \mathbf{n}_a \triangleleft \rfloor \end{bmatrix}}_{\mathbf{A}_{cal}} \begin{bmatrix} {}^{B}\boldsymbol{\omega}_{B/G} \\ {}^{B}\mathbf{a} \end{bmatrix}$$

$$\cdots + \underbrace{\begin{bmatrix} \mathbf{b}_\omega + \mathbf{c}_{\omega_1} + \mathbf{c}_{\omega_2} \\ \mathbf{b}_a + \mathbf{c}_{a_1} + \mathbf{c}_{a_2} \end{bmatrix}}_{\mathbf{b}_{cal}} + \underbrace{\begin{bmatrix} \frac{\rho_\omega}{\sqrt{2}} \mathbf{w}_\omega \\ \frac{\rho_a}{\sqrt{2}} \mathbf{w}_a \end{bmatrix}}_{\mathbf{w}_{imu}}$$

The notational conventions are listed at the end of the report, and ρ_ω and ρ_a are the noise densities for the gyroscopes and accelerometers, respectively. Technically, the definitions for misalignment and non-orthogonality are only approximations and not true rotations. However, this approximation is sufficient when the actual value of these rotations is not of interest. Extracting the true specific force and rotation rates leads to:

$$\begin{bmatrix} {}^{B}\boldsymbol{\omega}_{B/G} \\ {}^{B}\mathbf{a} \end{bmatrix} = \mathbf{A}_{cal}^{-1} \left(\begin{bmatrix} {}^{B}\breve{\boldsymbol{\omega}}_{B/G} \\ {}^{B}\breve{\mathbf{a}} \end{bmatrix} - \mathbf{b}_{cal} - \mathbf{w}_{imu} \right).$$

The estimated angular rates and specific forces are outputs of the system that are used to perform the integration:

$$\begin{bmatrix} {}^{B}\hat{\boldsymbol{\omega}}_{B/G} \\ {}^{B}\hat{\mathbf{a}} \end{bmatrix} = \hat{\mathbf{A}}_{cal}^{-1} \left(\begin{bmatrix} {}^{B}\breve{\boldsymbol{\omega}}_{B/G} \\ {}^{B}\breve{\mathbf{a}} \end{bmatrix} - \hat{\mathbf{b}}_{cal} \right).$$

The errors are additive:

$${}^{B}\tilde{\boldsymbol{\omega}}_{B/G} = {}^{B}\boldsymbol{\omega}_{B/G} - {}^{B}\hat{\boldsymbol{\omega}}_{B/G}, \qquad {}^{B}\tilde{\mathbf{a}} = {}^{B}\mathbf{a} - {}^{B}\hat{\mathbf{a}}.$$

The errors get quite complicated if terms like misaligment and g-sensitivity are actively being estimated. For implementation, extensive use of the MATLAB symbolic toolbox[15] was used to generate functions for the system outputs, Jacobians, and propagation matrices.

The correlated noise is meant to approximate "flicker" noise, which is used to model bias instability. The bias instability σ_{BI} of a sensor corresponds to a flat region on sensor noise Allan deviation (AD) curve at the level $\sigma_{AD} = \sigma_{BI}\sqrt{2\ln(2)/\pi}$.[16] Since flicker is a nonlinear stochastic process, it is approximated by two exponentially correlated noise processes. The analytical expression for the AD resulting from these processes is given in IEEE standards,[17] and a Levenberg-Marquadt optimization routine was used to adjust the τ_C and σ_C parameters of the two processes to minimize the error between the ideal and approximated AD curves. More exponentially correlated processes could be used to achieve a better approximation. An example of the resulting AD curve including the angle random walk, rate random walk, and two exponentially correlated noise processes is shown in Fig. 2.

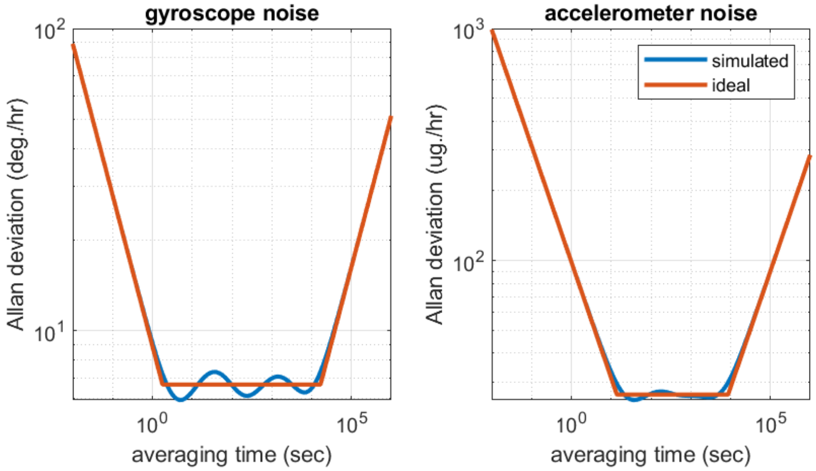


Fig. 2 AD curves representing how the sensor noise is modeled. The ideal curve consists of angle random walk, bias instability, and rate random walk contributions. The approximated curve replaces the bias-instability process with two exponentially correlated noise processes.

Table 2 IMU dynamic system states and how they are modeled in the navigation filter

Name	$\tilde{\mathbf{x}} \oplus \hat{\boldsymbol{x}}$	$\dot{\boldsymbol{x}} = f(\dots)$	$\dot{\tilde{\mathbf{x}}} = \tilde{f}(\dots)$
attitude	$e^{-\lfloor \tilde{\boldsymbol{\theta}} \times \rfloor}\,{}^B\hat{\mathbf{R}}_G$	${}^B\dot{\mathbf{R}}_G = -\left\lfloor {}^B\boldsymbol{\omega}_{B/G} \times \right\rfloor {}^B\mathbf{R}_G$	$\dot{\tilde{\boldsymbol{\theta}}} \approx -\left\lfloor {}^B\hat{\boldsymbol{\omega}}_{B/G} \times \right\rfloor \tilde{\boldsymbol{\theta}} + {}^B\tilde{\boldsymbol{\omega}}_{B/G}$
velocity	${}^G\tilde{\mathbf{v}}_B + {}^G\hat{\mathbf{v}}_B$	${}^G\dot{\mathbf{v}}_B = {}^B\mathbf{R}_G^{\top}\,{}^B\mathbf{a} + {}^G\mathbf{g}$	${}^G\dot{\tilde{\mathbf{v}}}_B \approx {}^B\hat{\mathbf{R}}_G^{\top}\,{}^B\tilde{\mathbf{a}} - {}^B\hat{\mathbf{R}}_G^{\top} \left\lfloor {}^B\hat{\mathbf{a}} \times \right\rfloor \tilde{\boldsymbol{\theta}}_B$
position	${}^G\tilde{\mathbf{p}}_B + {}^G\hat{\mathbf{p}}_B$	${}^G\dot{\mathbf{p}}_B = {}^G\mathbf{v}_B$	${}^G\dot{\tilde{\mathbf{p}}}_B = {}^G\tilde{\mathbf{v}}_B$
gyroscope bias	$\tilde{\mathbf{b}}_\omega + \hat{\mathbf{b}}_\omega$	$\dot{\mathbf{b}}_\omega = \sigma_{b_\omega} \mathbf{w}_{b_\omega}$	$\dot{\tilde{\mathbf{b}}}_\omega = \sigma_{b_\omega} \mathbf{w}_{b_\omega}$
accelerometer bias	$\tilde{\mathbf{b}}_a + \hat{\mathbf{b}}_a$	$\dot{\mathbf{b}}_a = \sigma_{b_a} \mathbf{w}_{b_a}$	$\dot{\tilde{\mathbf{b}}}_a = \sigma_{b_a} \mathbf{w}_{b_a}$
gyro. correlated noise process i	$\tilde{\mathbf{c}}_{\omega_i} + \hat{\mathbf{c}}_{\omega_i}$	$\dot{\mathbf{c}}_{\omega_i} = -\frac{1}{\tau_{c_{\omega_i}}} \mathbf{c}_{\omega_i} + \sigma_{c_{\omega_i}} \mathbf{w}_{c_{\omega_i}}$	$\dot{\tilde{\mathbf{c}}}_{\omega_i} = -\frac{1}{\tau_{c_{\omega_i}}} \tilde{\mathbf{c}}_{\omega_i} + \sigma_{c_{\omega_i}} \mathbf{w}_{c_{\omega_i}}$
accel. correlated noise process i	$\tilde{\mathbf{c}}_{a_i} + \hat{\mathbf{c}}_{a_i}$	$\dot{\mathbf{c}}_{a_i} = -\frac{1}{\tau_{c_{a_i}}} \mathbf{c}_{a_i} + \sigma_{c_{a_i}} \mathbf{w}_{c_{a_i}}$	$\dot{\tilde{\mathbf{c}}}_{a_i} = -\frac{1}{\tau_{c_{a_i}}} \tilde{\mathbf{c}}_{a_i} + \sigma_{c_{a_i}} \mathbf{w}_{c_{a_i}}$
gyro. scale adjustment	$\tilde{\mathbf{s}}_\omega + \hat{\mathbf{s}}_\omega$	$\dot{\mathbf{s}}_\omega = \mathbf{0}$	$\dot{\tilde{\mathbf{s}}}_\omega = \mathbf{0}$
accel. scale adjustment	$\tilde{\mathbf{s}}_a + \hat{\mathbf{s}}_a$	$\dot{\mathbf{s}}_a = \mathbf{0}$	$\dot{\tilde{\mathbf{s}}}_a = \mathbf{0}$
gyro. misalignment	$\tilde{\mathbf{m}}_\omega + \hat{\mathbf{m}}_\omega$	$\dot{\mathbf{m}}_\omega = \mathbf{0}$	$\dot{\tilde{\mathbf{m}}}_\omega = \mathbf{0}$
accel. misalignment	$\tilde{\mathbf{m}}_a + \hat{\mathbf{m}}_a$	$\dot{\mathbf{m}}_a = \mathbf{0}$	$\dot{\tilde{\mathbf{m}}}_a = \mathbf{0}$
gyro. non-orthogonality	$\tilde{\mathbf{n}}_\omega + \hat{\mathbf{n}}_\omega$	$\dot{\mathbf{n}}_\omega = \mathbf{0}$	$\dot{\tilde{\mathbf{n}}}_\omega = \mathbf{0}$
accel. non-orthogonality	$\tilde{\mathbf{n}}_a + \hat{\mathbf{n}}_a$	$\dot{\mathbf{n}}_a = \mathbf{0}$	$\dot{\tilde{\mathbf{n}}}_a = \mathbf{0}$
gyro. g-sensitivity	$\tilde{\mathbf{g}}_\omega + \hat{\mathbf{g}}_\omega$	$\dot{\mathbf{g}}_\omega = \mathbf{0}$	$\dot{\tilde{\mathbf{g}}}_\omega = \mathbf{0}$

5.1.2 System Partitioning

All versions of the system will include the position, attitude, and velocity error states, but there is variability in which of the calibration error states are modeled and how. There are three main variants, which are summarized in Table 3. The *oversimplified* variant does not estimate or explicitly model any of the higher order terms. Instead, covariance inflation factors must be determined in order to make this filter consistent. The advantage of this filter is speed and ease of implementation. The *optimal* variant does not model anything as a consider state; it models the full system and tries to actively estimate all of the error states. The advantage of this filter is that it should give the best performance; at least if there is sufficient motion for all of the error states to be observable. The disadvantage is that it has a significantly higher number of matrix operations in the Kalman update step, and may be unstable if the system is unobservable. The *balanced* variant is a suboptimal filter that still models the full system. Biases are treated as active states, but all of the higher order terms are modeled as consider states.

Table 3 State vectors for each state in each IMU system variant using the convention in Fig. 1

Error State	Oversimplified	Optimal	Balanced
attitude	a	a	a
velocity	a	a	a
position	a	a	a
gyro./accel. bias	b	b	b
gyro./accel. correlated noise processes		a	c
gyro./accel. scale adjustment		b	d
gyro./accel. misalignment		b	d
gyro./accel. non-orthogonality		b	d
gyro. g-sensitivity		b	d

5.2 Magnetometers

By sensing the Earth's magnetic field in body coordinates, magnetometers provide orientation information that is also very beneficial to estimating gyroscope biases. Magnetometers come with their own errors that need to be modeled.

5.2.1 System Modeling

The states in the magnetometer system are listed in Table 4. It is assumed that the Earth's magnetic field in global coordinates ${}^G\mathbf{m}$ is known. This is the only external input to the system. The model of the magnetometer sensor output ${}^B\breve{\mathbf{m}}$ includes calibration errors as well as attitude errors. In this case, the raw magnetometer output (${}^B\breve{\mathbf{m}}$) is modeled as a function of the true magnetic field in body-coordinates

Table 4 Magnetometer system states and how they are modeled

Name	$\tilde{\mathbf{x}} \oplus \hat{\boldsymbol{x}}$	$\dot{\boldsymbol{x}} = f(\dots)$	$\dot{\tilde{\mathbf{x}}} = \tilde{f}(\dots)$
magnetometer bias	$\tilde{\mathbf{b}}_m + \hat{\mathbf{b}}_m$	$\dot{\mathbf{b}}_m = \mathbf{0}$	$\dot{\tilde{\mathbf{b}}}_m = \mathbf{0}$
magnetometer soft-iron adjustments	$\tilde{\mathbf{s}}_m + \hat{\mathbf{s}}_m$	$\dot{\mathbf{s}}_m = \mathbf{0}$	$\dot{\tilde{\mathbf{s}}}_m = \mathbf{0}$
attitude (external)	$e^{-\lfloor\tilde{\boldsymbol{\theta}}\times\rfloor}\,{}^B\hat{\mathbf{R}}_G$	N/A	N/A

($^B\mathbf{m} = {}^B\mathbf{R}_G {}^G\mathbf{m}$) corrupted by errors:

$$\underbrace{{}^B\breve{\mathbf{m}}}_{\boldsymbol{z}_{mag}} = \underbrace{{}^B\mathbf{R}_G {}^G\mathbf{m} + \mathbf{b}_m + \lfloor \mathbf{s}_m \equiv \rfloor \, {}^B\mathbf{R}_G {}^G\mathbf{m} + \sigma_m \mathbf{v}_m}_{h_{mag}(\boldsymbol{x},\mathbf{e},\mathbf{u},\mathbf{v})}, \tag{23}$$

where $\mathbf{v}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Magnetometers that are surrounded by materials with different magnetic permeabilities are subjected to "soft-iron" errors, caused by the magnetic field being attenuated as a function of the platform orientation. In addition, proximity to magnetized objects on the same platform create constant "hard-iron" errors. These can be estimated in calibration,[18] but the lumped errors in the calibration will typically dominate normal sensor errors such as sensitivity error or electrical bias. Therefore, all of the misalignment, cross-axis sensitivity, and nonorthogonality errors have been lumped into a "soft-iron" matrix parameterized by the 9×1 vector $\mathbf{s}_m$, and the biases and hard-iron errors have been lumped into $\mathbf{b}_m$. The predictions are given by:

$$\underbrace{{}^B\hat{\breve{\mathbf{m}}}}_{\hat{\boldsymbol{z}}_{mag}} = \underbrace{{}^B\hat{\mathbf{R}}_G {}^G\mathbf{m} + \hat{\mathbf{b}}_m + \lfloor \hat{\mathbf{s}}_m \equiv \rfloor \, {}^B\hat{\mathbf{R}}_G {}^G\mathbf{m}}_{\hat{h}_{mag}(\hat{\boldsymbol{x}},\mathbf{u},\hat{\mathbf{e}})}, \tag{24}$$

If the error definitions from Table 4 are used in Eq. 23, the residual equation can be determined by:

$$\tilde{h}_{mag}\left(\hat{\boldsymbol{x}}, \tilde{\mathbf{x}}, \mathbf{u}, \hat{\mathbf{e}}, \tilde{\mathbf{e}}, \mathbf{v}\right) = \boldsymbol{z}_{mag} - \hat{\boldsymbol{z}}_{mag} \tag{25}$$

It is sufficient to use the first order approximation to the matrix exponential for attitude errors, i.e., ${}^B\mathbf{R}_G \approx \left(\mathbf{I} - \left\lfloor \tilde{\boldsymbol{\theta}} \times \right\rfloor\right) {}^B\hat{\mathbf{R}}_G$. The Jacobians in Eq. 9 can then be computed manually or with symbolic math software.

5.2.2 System Partitioning

Like the IMU system, the magnetometer also has an *oversimplified*, *optimal*, and *balanced* variant. The partitioning assignments are summarized in Table 5. For the magnetomer system, the soft-iron effects are neglected in the *oversimplified* variant, treated as static-active states in the *optimal* variant, and treated as static-consider states in the *balanced* variant.

Table 5 State vectors for each state in each IMU system variant using the convention in Fig. 1

Error State	Oversimplified	Optimal	Balanced
magnetometer bias	b	b	b
magnetometer soft-iron adjustments		b	d

5.3 Black-Box GPS System

A GPS receiver provides position and velocity estimates if it can acquire enough satellites. The “black-box” terminology refers to using the position and velocity measurements produced by the receiver’s internal least-squares solution as opposed to using a more tightly coupled algorithm that combines the raw pseudorange and doppler shift measurements with the rest of the navigation system. A system can be constructed to model the GPS errors and the GPS outputs.

5.3.1 System Modeling

The GPS model consists of position and velocity measurements corrupted with correlated noise. The internal and external system states are listed in Table 6. In the models, $\mathbf{w}_p$ and $\mathbf{w}_v$ are unit white noise processes, β_p and β_v are bandwidths, and σ_p^2 and σ_v^2 are variances. The system outputs are the raw GPS measurements given

Table 6 GPS system states and how they are modeled

Name	$\tilde{\mathbf{x}} \oplus \hat{\boldsymbol{x}}$	$\dot{\boldsymbol{x}} = f(\dots)$	$\dot{\tilde{\mathbf{x}}} = \tilde{f}(\dots)$
correlated position bias	$\tilde{\mathbf{b}}_p + \hat{\mathbf{b}}_p$	$\dot{\mathbf{b}}_p = -\beta_p \mathbf{b}_p + \sqrt{2\sigma_p^2\beta_p}\mathbf{w}_p$	$\dot{\tilde{\mathbf{b}}}_p = -\beta_p \tilde{\mathbf{b}}_p + \sqrt{2\sigma_p^2\beta_p}\mathbf{w}_p$
correlated velocity bias	$\tilde{\mathbf{b}}_v + \hat{\mathbf{b}}_v$	$\dot{\mathbf{b}}_v = -\beta_v \mathbf{b}_v + \sqrt{2\sigma_v^2\beta_v}\mathbf{w}_v$	$\dot{\tilde{\mathbf{b}}}_v = -\beta_v \tilde{\mathbf{b}}_v + \sqrt{2\sigma_v^2\beta_v}\mathbf{w}_v$
position (external)	$\tilde{\mathbf{p}}_B + \hat{\mathbf{p}}$	N/A	N/A
velocity (external)	$\tilde{\mathbf{v}}_B + \hat{\mathbf{v}}$	N/A	N/A

by:

$$\underbrace{\begin{bmatrix} \breve{\mathbf{p}}_{GPS} \\ \breve{\mathbf{v}}_{GPS} \end{bmatrix}}_{\boldsymbol{z}_{gps}} = \underbrace{\begin{bmatrix} {}^{G}\mathbf{p}_B + \mathbf{b}_p \\ {}^{G}\mathbf{v}_B + \mathbf{b}_v \end{bmatrix}}_{h_{gps}(\mathbf{x},\mathbf{e})} . \tag{26}$$

There are no external inputs or noise, the system outputs depend on the states and external states. From here, computing the predictions, residuals, and Jacobians for the system outputs is trivial.

5.3.2 System Partitioning

Like the IMU system, the GPS system also has an *oversimplified*, *optimal*, and *balanced* variant. The partitioning assignments are summarized in Table 7. The correlated biases are neglected in the *oversimplified* variant. Instead, they are approximated by measurement noise. The correlated biases are treated as dynamic-active states in the *optimal* variant, and dynamic-consider states in the *balanced* variant.

Table 7 State vectors for each state in each GPS system variant using the convention in Fig. 1

Error State	Oversimplified	Optimal	Balanced
correlated position bias		a	c
correlated velocity bias		a	c

5.4 Simulation Setup

The various filter designs were evaluated in two separate dynamic platforms. The first simulation was of a "tumble test" calibration. Essentially, the IMU platform is held stationary while it is rotated into multiple orientations. There are 26 setpoint orientations with a dwell time of 5 s each. In order to not instantaneously jump to the different orientations, the IMU is assumed to be attached to a platform with non-negligible moments of inertia that must be rotated into position with a proportional controller. This produces settling times around 2 s and transient angular rates of under 100 deg/s. The second platform is a fictitious gun-launched projectile that deploys wings at apogee, then glides for several minutes to hit a target that is several kilometers downrange and to the right of the line of fire. The munition launches with some initial uncertainty that is used in the initialization of the IMU states. The initial condition variances shown in Table 8 are used to tune the initial covariance of the IMU system. Each platform includes the same sensor models. Example sensor outputs for each of the simulations are shown in Fig. 3.

The IMU specifications are listed in Table 9. The magnetometer and GPS specifications are listed in Table 10. Both the IMU and magnetometer errors are based on the VN-100 data sheet,[19] although some errors had to be guessed at using engineering judgement. The GPS parameters are based off of past experience fitting u-blox reciever errors to mortar trajectories run with a Spirent GPS simulator. To evaluate navigation drift in a GPS-denied environment, the GPS measurements were stopped one-third of the way through the simulation.

Table 8 Initial condition standard deviations

Name	**Munition**	**Calibration**
Launch velocity 1σ (m/s)	3	0
Launch elevation 1σ (deg)	0.02813	0
Launch azimuth 1σ (deg)	0.1519	0
Initial roll error 1σ (deg)	0.1	0

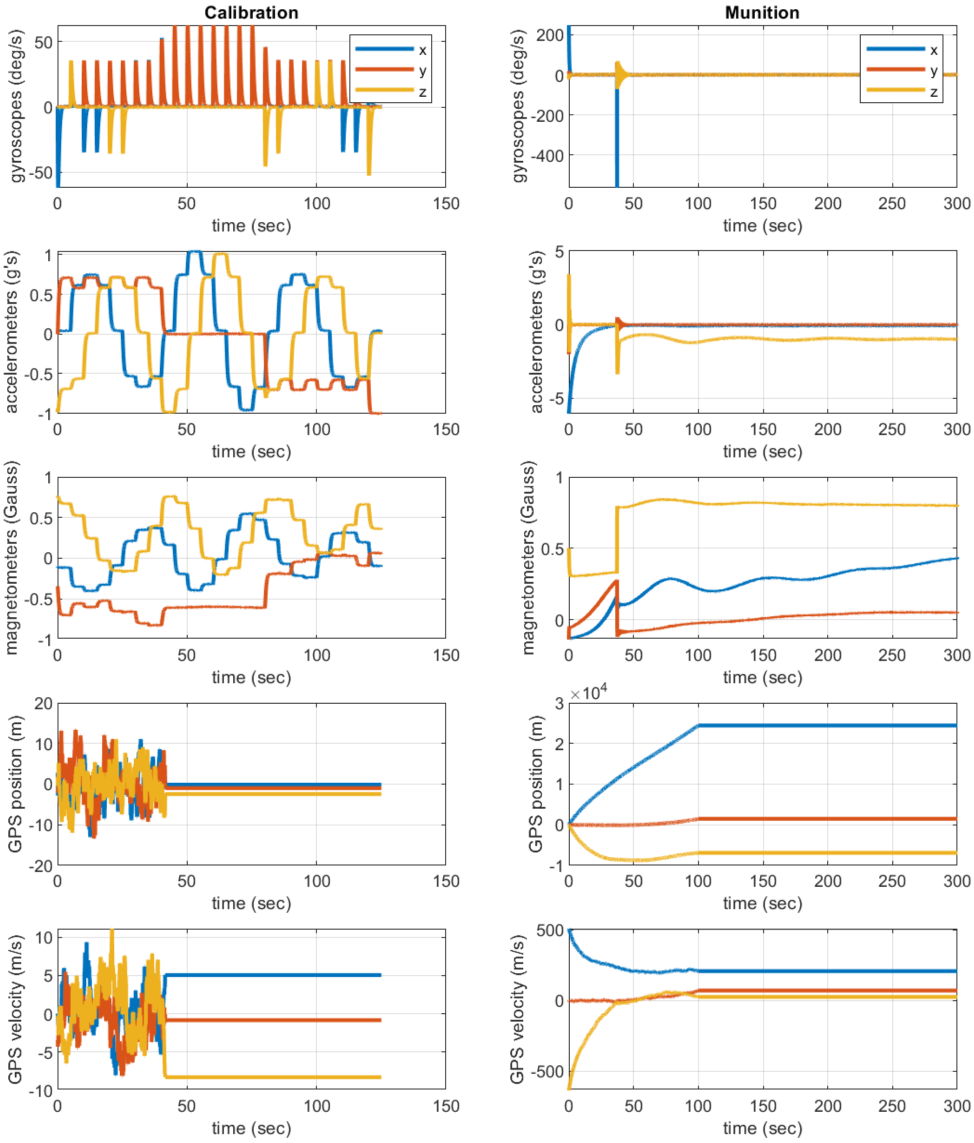


Fig. 3 Example sensor outputs from the two sample trajectories. Notice that the GPS stops providing outputs one-third of the way through each trajectory.

Table 9 IMU simulation parameters

Gyroscope		Accelerometer	
Name	Value	Name	Value
Sample rate (Hz)	500	Sample rate (Hz)	500
Noise density (deg/sec/rt-Hz))	0.0035	Bias stability (mg)	0.04
Bias stability (deg/hr)	10	Noise density (mg/sqrt(Hz))	0.14
Rate random walk (deg/sec/rt-hr)	0.001485	Acceleration random walk (mg/sec/rt-hr)	0.0297
Bias repeatability 1σ (deg/sec)	0.5	Bias repeatability 1σ (mg)	20
Sensitivity repeatability 1σ (%)	0.3333	Sensitivity repeatability 1σ (%)	0.3333
Misalignment 1σ (deg)	0.05	Misalignment 1σ (deg)	0.05
Non-orthogonality 1σ (deg)	0.05	Non-orthogonality 1σ (deg)	0.05
g-Sensitivity 1σ (deg/sec/g)	0.015	Nonlinearity 1σ (%FSV)	0
Nonlinearity 1σ (%FSV)	0		

Table 10 Magnetometer and GPS simulation parameters

Magnetometer		GPS	
Name	Value	Name	Value
Sample rate (Hz)	50	Sample rate (Hz)	10
Nonlinearity (%FSV)	0	Position β_p (Hz)	1
Bias repeatability (mGauss)	610	Position σ_p (m)	5
Noise density (mGauss/sqrt(Hz))	0.0259	Velocity β_v (Hz)	0.4
Soft iron errors (%)	1	Velocity σ_v (m/s)	4

5.4.1 Initial Results

Small Monte-Carlo batches of 20 runs were performed with each filter and trajectory combination. Two error metrics are used to measure filter performance: root sum of squared error (RSSE) and normalized estimation error squared (NEES), which are defined as follows:

$$\begin{aligned} RSSE\left(\tilde{\mathbf{x}}\right) &= \sqrt{\tilde{\mathbf{x}}^{\top}\tilde{\mathbf{x}}} \\ NEES\left(\tilde{\mathbf{x}}, \mathbf{P}_{xx}\right) &= \tilde{\mathbf{x}}^{\top}\mathbf{P}_{xx}^{-1}\tilde{\mathbf{x}} \end{aligned}$$

RSSE is essentially the norm of the state error x that is computed from the true state provided by the simulation and the estimated state using the appropriate $\ominus$ operator. NEES takes into account $\mathbf{P}_{xx}$, the block of the estimator covariance associated with x to compute the Mahalanobis distance. The Mahalanobis distance follows a χ^2 distribution, and provides a measure of how closely the true state errors fit into the distribution predicted by the EKF. These measurements are performed on subsections of the state vector: attitude, velocity, and position. Because each of these subsections have 3 degrees of freedom, a NEES of 3 indicates the estimator is consistent. Both metrics are computed for each run in the Monte-Carlo simulation, and then averaged together. The RSSE results are contained in Fig. 4, and the NEES results are in Fig. 5.

Two trends are apparent from the data. The first is that the *oversimplified* filter is significantly less accurate and less consistent than the other estimators. It is possible that additional tuning of the covariance inflations could fix this, but this tuning is neither simple, nor robust. Too much covariance inflation causes large attitude errors, which start to violate the small-angle approximations used to propagate the velocity and position errors. Too little covariance inflation causes the filter to overweight either the process model or the measurements. The second trend is that while both the *optimal* and *balanced* filter are consistent (they both have a NEES of about 3), the *balanced* estimator has about twice as much RSSE error on average when GPS is removed.

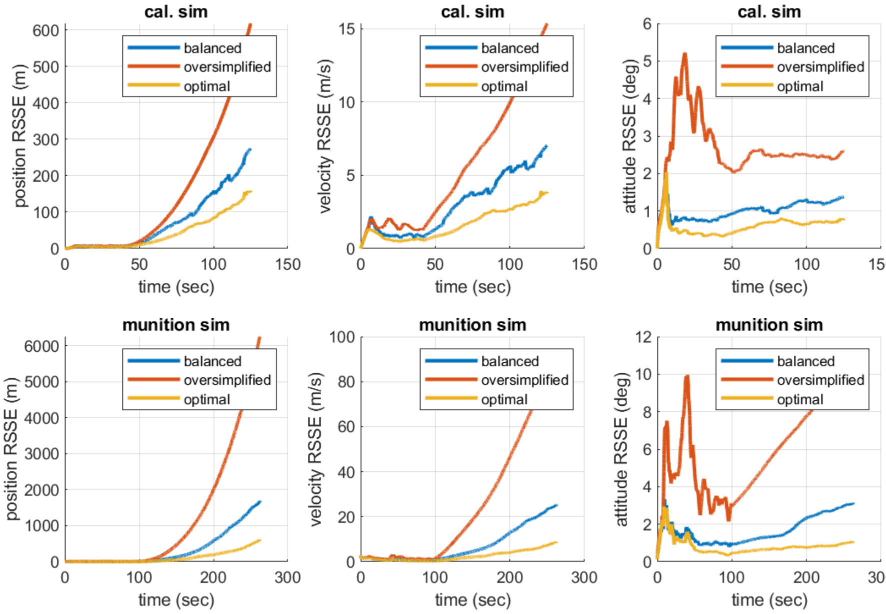


Fig. 4 Average RSSE results for each filter on each trajectory

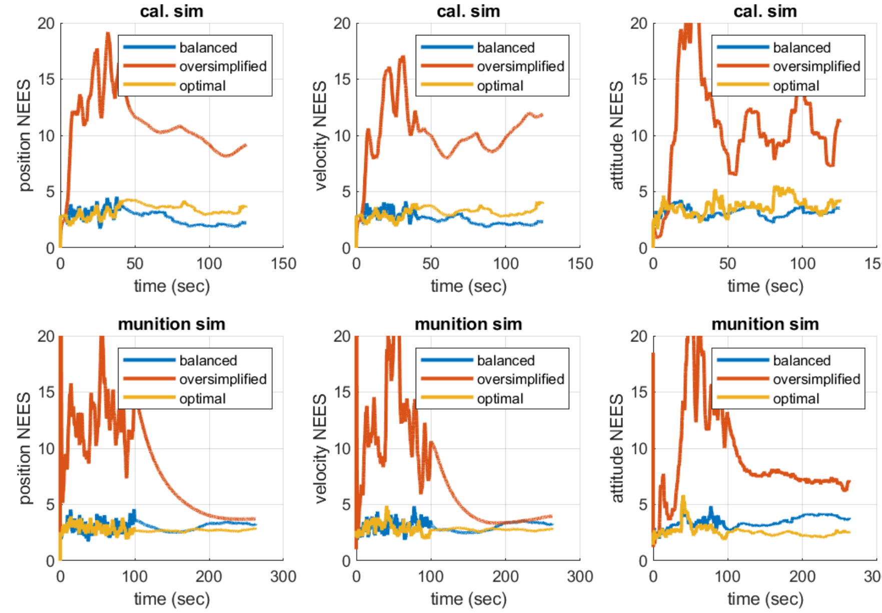


Fig. 5 Average NEES results for each filter on each trajectory

5.4.2 Error Budget

The *balanced* filter is less accurate than the *optimal* filter, and uses consider states extensively. An error budget is a useful tool that estimates how much each error source contributes to the total variance of the estimator. It can also be used to decide if it makes sense to change the filter to estimate any of the error states that are currently modeled as consider states. Error budget computation has been described in Gelb,[4] Estefan et. al.,[20] and more recently by Geller et. al.[21] To compute the error budget, the systems that build up the Kalman filter are duplicated in a "truth model". The filter model error states and covariance $\tilde{\mathbf{x}}_f$, $\mathbf{P}_{ff}$ and the truth model error states and covariance $\tilde{\mathbf{x}}_t$, $\mathbf{P}_{tt}$ form the augmented system:

$$\begin{aligned}
\tilde{\mathbf{x}}_a &= \begin{bmatrix} \tilde{\mathbf{x}}_f \\ \tilde{\mathbf{x}}_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \left[\begin{bmatrix} \mathbf{P}_{ff} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{tt} \end{bmatrix} \right] \right) \\
\begin{bmatrix} \dot{\tilde{\mathbf{x}}}_f \\ \dot{\tilde{\mathbf{x}}}_t \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_f \\ \tilde{\mathbf{x}}_t \end{bmatrix} + \begin{bmatrix} \mathbf{G}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_t \end{bmatrix} \begin{bmatrix} \mathbf{w}_f \\ \mathbf{w}_t \end{bmatrix} \\
\begin{bmatrix} \tilde{\mathbf{z}}_f \\ \tilde{\mathbf{z}}_t \end{bmatrix} &= \begin{bmatrix} \mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_f \\ \tilde{\mathbf{x}}_t \end{bmatrix} + \begin{bmatrix} \mathbf{L}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_t \end{bmatrix} \begin{bmatrix} \mathbf{v}_f \\ \mathbf{v}_t \end{bmatrix}
\end{aligned}$$

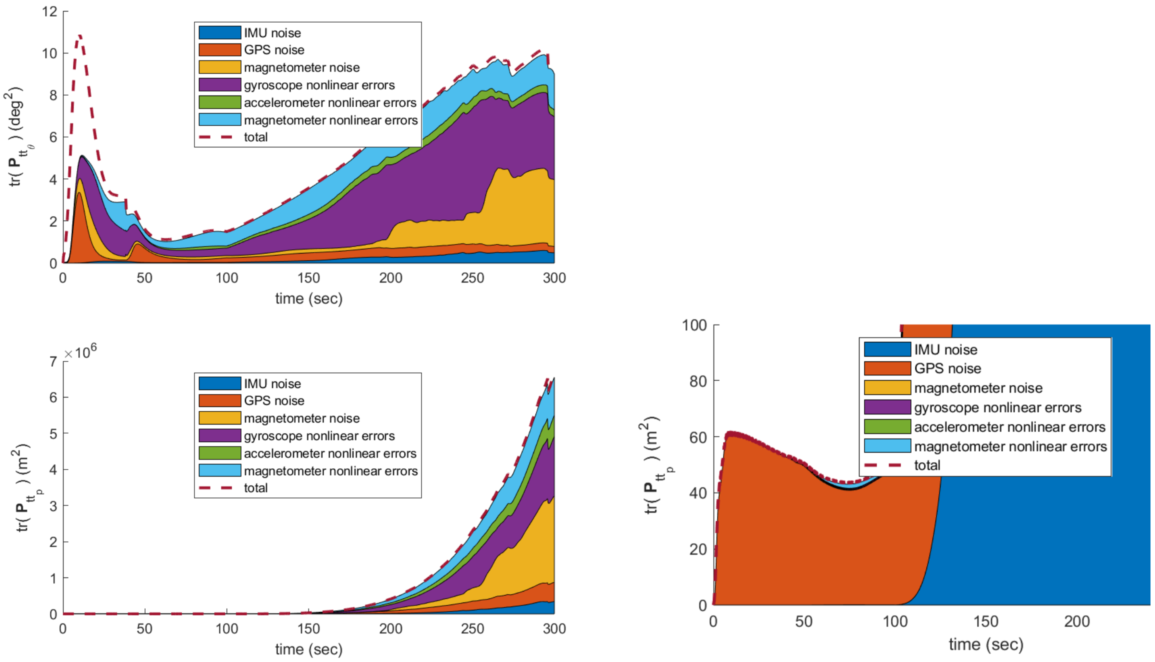
To determine the variance contribution from a specific error source (or group of error sources), the truth model is tuned with every element in $\mathbf{L}_t$, $\mathbf{G}_t$, and the initial value of $\mathbf{P}_{tt}$ set to 0 except for the elements relating to that error source. The filter model tuning never changes. The two models are run side-by-side and share all of the same linearization points. That is, a state vector from a nominal trajectory is used for all linearization, and the system model only propagates and updates the covariance. However, when a measurement is used, the gain is computed from the filter model, and the same gain is used for both systems:

$$\begin{aligned}
\mathbf{K}_f &= \mathbf{P}_{ff}\mathbf{H}^\top \left(\mathbf{H}\mathbf{P}_{ff}\mathbf{H}^\top + \mathbf{L}\mathbf{L}^\top \right)^{-1} \\
\begin{bmatrix} \mathbf{P}_{ff} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{tt} \end{bmatrix}^+ &= \begin{bmatrix} \mathbf{I} - \mathbf{K}_f\mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathbf{K}_f\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{ff} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{tt} \end{bmatrix} \begin{bmatrix} \mathbf{I} - \mathbf{K}_f\mathbf{H} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - \mathbf{K}_f\mathbf{H} \end{bmatrix}^\top \\
&\quad + \begin{bmatrix} \mathbf{K}_f\mathbf{L}_f\mathbf{L}_f^\top\mathbf{K}_f^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_f\mathbf{L}_t\mathbf{L}_t^\top\mathbf{K}_f^\top \end{bmatrix}
\end{aligned}$$

Note that in this case the filter and truth models remain uncorrelated. For this reason, it is common to run the filter model first, save the Kalman gains, and then run the truth models with the saved Kalman gains.

An initial error budget was performed for the *balanced* filter, on the munition trajectory. The resulting error contributions are shown in Fig. 6. The figure shows the traces of two blocks of the truth model covariance matrix: the attitude error block $\mathbf{P}_{tt_\theta}$ and the position error block $\mathbf{P}_{tt_p}$. The error categories were purposely broad: all IMU noise, all GPS noise, all magnetometer noise, all of the gyroscope nonlinear errors, all of the accelerometer nonlinear errors, and all of the magnetometer nonlinear errors. Notice that the total variance is also shown on the plots. Because the filter model and truth model are always linearized around the same state, and the covariance operations are linear, all of the individual error variances should add up to the total error variance. Early on, the categories examined do not sum up to the total error variance for attitude. This is because the categories do not include all of the errors that drive the system. In particular, the categories do not include initial condition variances, or initial sensor bias variances (since biases are all estimated by the balanced filter). Because the filter model is observable while there is GPS, the variance contribution from these errors becomes negligible after about 40 s.

After GPS is removed, the three largest contributors to the position error are the magnetometer noise, the gyroscope nonlinear errors, and the magnetometer nonlinear errors. The errors from magnetometer noise cannot be removed by an EKF, but it is possible that if some of the gyroscope or magnetometer nonlinear errors were modeled as active states instead of consider states the filter performance would improve. However, this adds computational complexity to the Kalman update operation, so model changes should be reserved for the errors that have the largest effect on performance. To determine this, a second error budget was created that “zooms in” on the gyroscope and magnetometer nonlinear errors. Instead of turning groups of sensor errors on at a time, specific nonlinear errors were turned on individually. The results of this error budget are shown in Fig. 7. From the plots, gyroscope g-sensitivity and magnetometer soft-iron (these are the only nonlinear magnetometer errors) errors appear to dominate.



(a)

(b)

Fig. 6 Initial error budget: a) major error contributions; b) prior to GPS loss, position error is dominated by GPS noise

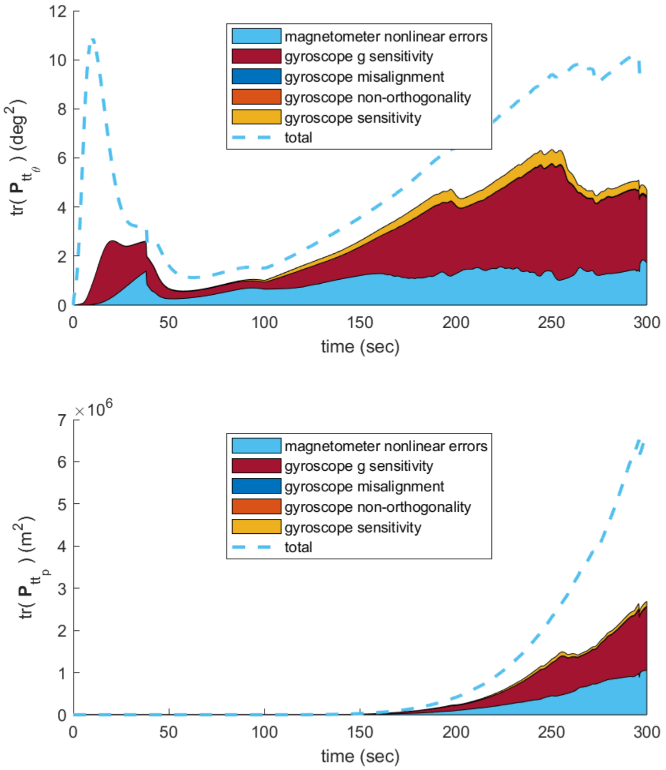


Fig. 7 Second error budget only looking at nonlinear gyroscope and magnetometer errors

5.4.3 Balanced Filter 2

From the error budget analysis, it was determined that magnetometer soft-iron effects and gyroscope g-sensitivity errors are the primary contributors to position drift after GPS-loss in the munition trajectory (using the error levels in this study). Magnetometer soft-iron errors are already treated as static active states in the *optimal* version of that system. A new IMU system was created called the *balanced 2* version that is identical to the *balanced* version with the exception that gyro g-sensitivity is modeled as a static-active state instead of a static-consider state. The *balanced 2* filter consists of the *balanced 2* IMU system, the *optimal* magnetometer system, and the *balanced* GPS system.

The Monte-Carlo analysis was repeated with all versions of the filter. The RSSE results are shown in Fig. 8, and the NEES results are shown in Fig. 9. As hoped for, the *balanced 2* filter has RSSE values that are much closer to the *optimal* RSSE values for the munition simulation. The RSSE values are not quite as close to the *optimal* RSSE values for the calibration simulation, but still better than the original *balanced* filter. It is intuitive that more errors are observable with the calibration simulation, which is why the *optimal* filter still outperforms both *balanced* filters. From the NEES data, the consistency of the *balanced 2* filter appears to be not significantly different than that of the *balanced* and *optimal* filters.

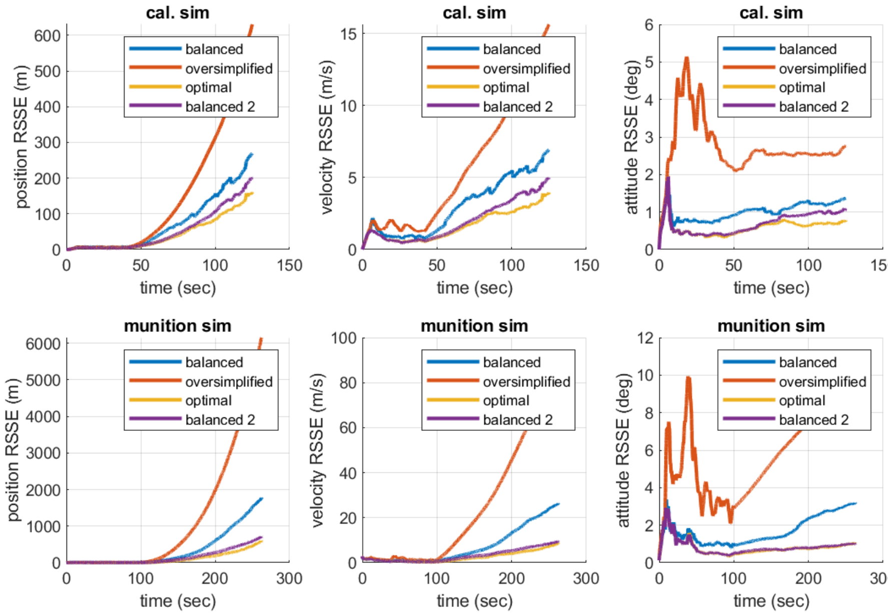


Fig. 8 Average RSSE results for each filter on each trajectory

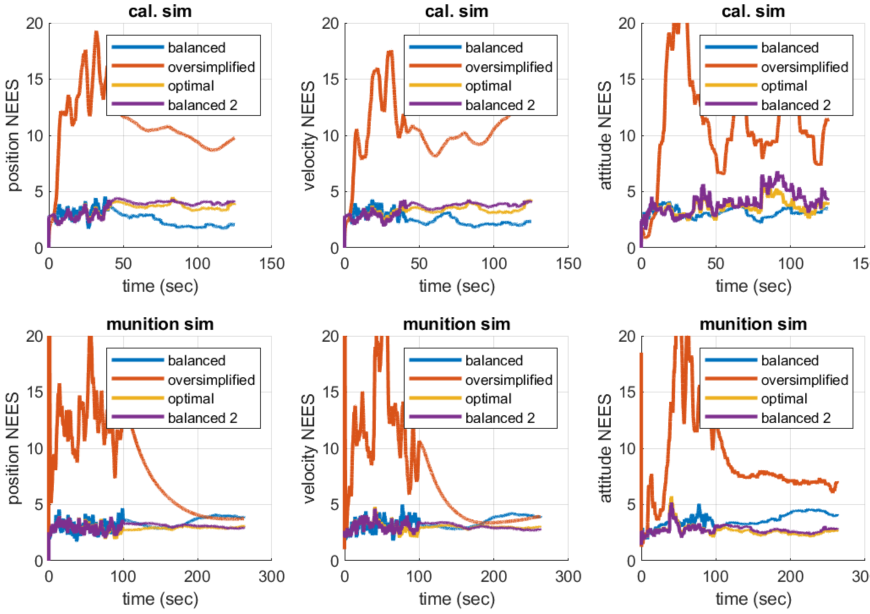


Fig. 9 Average NEES results for each filter on each trajectory

5.4.4 Performance Evaluation

One benefit to using consider states is a reduction in computational complexity. A crude comparison of the four filter versions was performed using MATLAB's profiling tools. The average times per function call using a 3.1 GHz processor are shown in Fig. 10. For comparison, the theoretical complexity as described in Section 3.3.2 for each filter is calculated and shown in Fig. 11. The MATLAB execution obviously has some overhead, since the average execution times for the *oversimplified* filter were similar to the other filters, when according to the complexity analysis they should be at least an order of magnitude less. The *optimal* filter had longer execution times than either *balanced* filter, which is consistent with the complexity analysis. It is unclear why the *balanced* filter took longer on average to compute the covariance propagation than the *balanced 2* filter.

For reference, the theoretical complexity plots also include the complexity of a completely unpartitioned version of the *optimal* filter, which is called the *naive full* filter. Because of the dynamic/static partitioning in the *optimal* and *balanced* filters, they have around 4 times less complexity than the *naive full* version when propagating the error covariance. The effect is less dramatic for performing the Kalman update. As expected, the *optimal* filter (which has no consider states) has the same complexity as the *naive full* filter for the Kalman update. The *naive full* filter has 60% higher Kalman update complexity than the *balanced* filter, and 28% higher Kalman update complexity than the *balanced 2* filter.

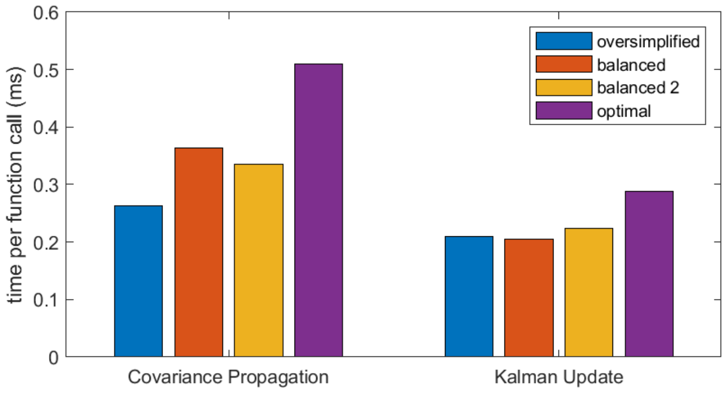


Fig. 10 Profiling results

The efficiency gains obtained by partitioning depend on the application. In this case,

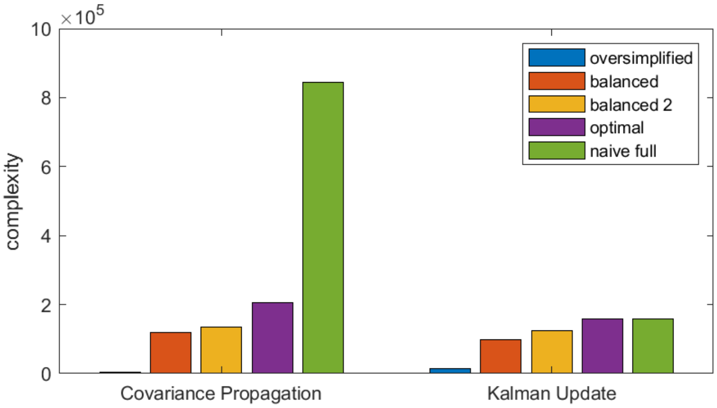


Fig. 11 Theoretical complexity

the profiling differences do not appear dramatic. However, suppose an embedded processor runs about 4 times slower than the computer used in this evaluation. The combined propagation and update execution time of about 4×0.58 ms and 4×0.80 ms for the *balanced 2* and *optimal* filters, respectively, mean the *balanced 2* filter can run at 431 Hz while the *optimal* filter will only run at 312 Hz. Having the ability to increase the update rate by 25% just by changing filter design could be useful when considered against the other guidance and control requirements of the vehicle.

5.4.5 Best-Case Error Contributions

So far, it has been demonstrated that if a consider state contributes significantly to the RSSE of the filter, then it can be remodeled as an active state to significantly reduce that RSSE contribution. Covariance analysis is a good way to model the augmented system to find a good balance between efficiency and RSSE performance. But what if the optimal filter was used, which would be the best-case scenario in terms of RSSE? There will still be errors in the state estimates, but at this point nothing else can be fixed with a modeling change. However, performing an error budget on the optimal filter might show which error sources could be nullified by choosing higher quality sensors. Such an error budget was performed on the munition trajectory, and the results are shown in Fig. 12. The error contributions to the attitude variance and position variance were ranked at 100 s (right before GPS loss), and at 295 s. Only the top 5 (out of 22) contributors are shown for each scenario. From the error budget, it would appear that magnetometer noise is the

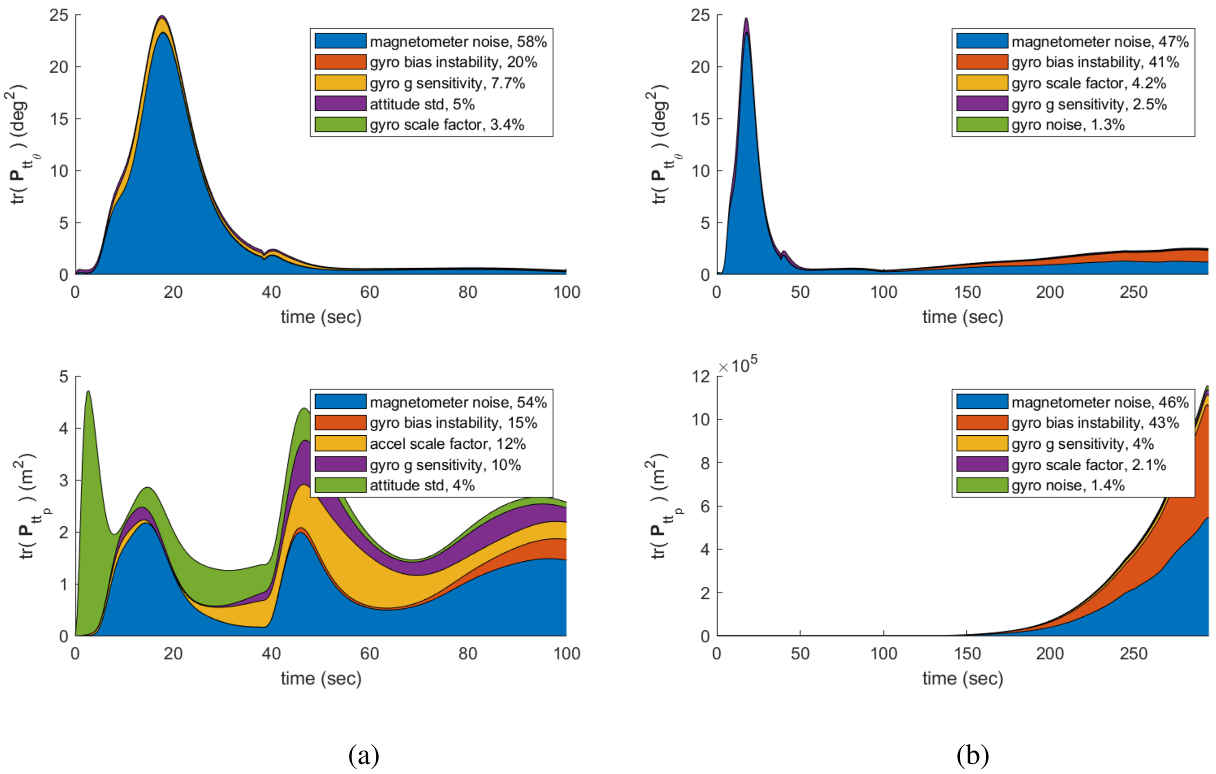


Fig. 12 Optimal filter error budget: a) contributions prior to GPS loss; b) contributions for whole trajectory. Legends list the top 5 contributors and their percentage of the total variance.

primary contributor to attitude and position error in all cases. Low attitude error is a necessary, but not sufficient, condition for low position error because specific forces must be transformed using the attitude estimate prior to integrating them. The next lowest contributor is gyroscope bias stability (or instability, depending on convention). This is a commonly listed specification for gyroscopes, with more stable gyroscopes typically costing more. Magnetometer noise and gyroscope stability are coupled, because observing magnetometers tends to make gyroscope bias observable. However, in practice magnetometers must contend with magnetic fields created by on-board electronics, which may result in larger magnetometer noise.

The position error budget prior to GPS loss is interesting. It would appear that by actively estimating the correlated GPS errors, their contribution to position errors has become negligible. Also, initial attitude errors and accelerometer scale factor error play a larger part than in the rest of the trajectory.

6. Conclusions

A modular error-state Kalman filtering approach has been developed. Using error-states allows the designer to perform filtering on manifolds (to a given level of approximation) while being able to compute Jacobians using symbolic math software. It was demonstrated that using a partitioned state vector can result in significant computational savings while still allowing the system to be correctly modeled. Fully modeling the system simplifies tuning because there are fewer (if any) “fudge factors” that need to be adjusted. It was also demonstrated that consider states can be used to improve filter efficiency, although no improvements to consistency were observed in the considered example. In the example, partitioning the filter along the lines of static and active states resulted in a dramatic reduction in complexity when performing the covariance propagation operation.

Error budget creation was discussed and used for two purposes with the munition trajectory. In one example, the error budget was used to determine which states should be remodeled as active states instead of consider states to achieve the best increase in filter accuracy. It was determined that estimating gyroscope g-sensitivity and magnetometer soft-iron errors caused a new suboptimal filter to perform nearly as well as the optimal filter on that trajectory. In the next example, an error budget was created for the optimal filter to show where performance was limited not by modeling decisions, but by sensor quality. The errors from sensor limitations were dominated by a combination of gyroscope instability and magnetometer noise.

The primary shortcoming of the method proposed here is the lack of external states in the propagation equations. Adding this would enable covariance analysis of navigation systems coupled with dynamic systems, such as in Geller et al.[21]

7. References

1. Simon D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches. Hoboken (NJ): Wiley-Interscience; 2006.

2. Brown RG, Hwang P. Introduction to random signals and applied Kalman filtering. (vol. 3) New York (NY): Wiley New York; 1997.

3. Maybeck PS. Stochastic models, estimation, and control. IEEE Transactions on Automatic Control. 1982;28(8): 868–869.

4. Gelb A. Applied optimal estimation. Cambridge (MA): MIT press; 1974.

5. Titterton D, Weston J. Strapdown inertial navigation technology. Second ed. London (UK): The Institution of Engineering and Technology; 2005.

6. Sola J, Deray J, Atchuthan D. A micro lie theory for state estimation in robotics. arXiv preprint arXiv:1812.01537. 2018.

7. The MathWorks I. Symbolic math toolbox. Natick (MA): MathWorks; 2019.

8. Trawny N, Roumeliotis SI. Indirect Kalman filter for 3D attitude estimation. Saint Paul (MN): University of Minnesota; 2005 Mar. Report No.: 2005-002.

9. Maley JM. Multiplicative quaternion extended Kalman filtering for nonspinning guided projectiles. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2013. Report No.: ARL-TR-6503.

10. Titterton D, Weston JL. Strapdown inertial navigation technology. Reston (VA): The American Institute of Aeronautics; 2004.

11. Simon D. Optimal state estimation. Hoboken (NJ): John Wiley and Sons, Inc.; 2006.

12. Geneva P, Maley J, Huang G. An efficient Schmidt-EKF for 3D visual-inertial slam. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; p. 12105–12115.

13. Mourikis AI, Roumeliotis SI, Burdick JW. SC-KF mobile robot localization: A stochastic cloning-Kalman filter for processing relative-state measurements. IEEE Transactions on Robotics. 2007;23(3):717-730.

14. Forster C, Carlone L, Dellaert F, Scaramuzza D. On-manifold preintegration for real-time visual–inertial odometry. IEEE Transactions on Robotics. 2017;33(1):1-21.

15. MATLAB release 2019b. Natick (MA): MathWorks; 2019.

16. El-Sheimy N, Hou H, Niu X. Analysis and modeling of inertial sensors using Allan variance. IEEE Transactions on Instrumentation and Measurement. 2008;57(1):140–149.

17. IEEE standard specification format guide and test procedure for single-axis laser gyros. IEEE Std 647-2006. 2006.

18. Ozyagcilar T. Calibrating an ecompass in the presence of hard and soft-iron interference. Freescale Semiconductor Ltd. 2012:1–17.

19. Vn-100 IMU/AHRS high-performance embedded navigation. Dallas, TX: VectorNav Technologies, LLC; 2016. 12-0002-R3.

20. Estefan JA, Burkhart PD. Sensitivity analysis of the enhanced orbit determination filter–error budget development. Washington (DC): NASA; 1994.

21. Geller DK, Christensen DP. Linear covariance analysis for powered lunar descent and landing. Journal of Spacecraft and Rockets. 2009;46(6):1231–1248.

List of Symbols, Acronyms, and Abbreviations

AD	Allan deviation
EKF	Extended Kalman Filter
GPS	global positioning system
IMU	inertial measurement unit
NEES	normalized estimation error squared
PSD	Power Spectral Density
RSSE	root sum of squared error

Notational Conventions

$\mathbf{v}$	Vectors are bold, roman font lowercase
$\mathbf{M}$	Matrices are bold uppercase
$\boldsymbol{x}$	Bold, italic, lowercase indicates a variable or set of variables that could be a vector, scalar, or matrix
${}^{C}\mathbf{v}_{B/A}$	A vector $\mathbf{v}$ describing something (e.g., velocity, position) of frame B with respect to frame A, represented in frame C coordinates
${}^{B}\mathbf{R}_{A}$	A rotation matrix that transforms a vector's A-frame coordinates to its B-frame coordinates (e.i. ${}^{B}\mathbf{v} = {}^{B}\mathbf{R}_{A}{}^{A}\mathbf{v}$)
$\lfloor \mathbf{v}\times \rfloor$	A skew-symmetric matrix constructed from vector $\mathbf{v}$ (e.g. $\lfloor \mathbf{v}\times \rfloor \mathbf{a} = \mathbf{v} \times \mathbf{a}$)
$\lfloor \mathbf{v}\triangleleft \rfloor$	A non-orthogonal error matrix formed from the vector $\lfloor \mathbf{v}\triangleleft \rfloor$ i.e. $\lfloor \mathbf{v}\triangleleft \rfloor = \begin{bmatrix} 0 & v_z & v_y \\ v_z & 0 & v_x \\ v_y & v_x & 0 \end{bmatrix}$
$\lfloor \mathbf{v}\backslash \rfloor$	A matrix with elements of the vector $\mathbf{v}$ along the diagonal
$\lfloor \mathbf{v} \equiv \rfloor$	A non-orthogonal error matrix formed from the vector $\mathbf{v} \in \mathbb{R}^9$ i.e. $\lfloor \mathbf{v} \equiv \rfloor = \begin{bmatrix} v_1 & v_4 & v_7 \\ v_2 & v_5 & v_8 \\ v_3 & v_6 & v_9 \end{bmatrix}$
$\mathbb{E}\left[\ldots\right]$	Expectation operator
$\lceil f \rceil_{\mathbf{v}}\Big\vert_{\mathbf{v}=\mathbf{y}}$	Jacobian of function f w.r.t. vector $\mathbf{v}$ evaluated at $\mathbf{v} = \mathbf{y}$
$\boldsymbol{x}$	The true value of variable $\boldsymbol{x}$
$\hat{\boldsymbol{x}}$	The estimated value of variable $\boldsymbol{x}$
$\tilde{\mathbf{x}}$	The error in $\hat{\boldsymbol{x}}$, i.e. $\boldsymbol{x} = e\left(\tilde{\mathbf{x}}, \hat{\boldsymbol{x}}\right)$ ($\tilde{\mathbf{x}}$ is always a vector)
$\breve{\mathbf{v}}$	The raw measured value of vector $\mathbf{v}$ from a sensor
$\breve{\mathbf{v}}^{-}$	The output vector $\mathbf{v}$ from a sensor predicted from the current state estimate
$\mathbf{v}'$	A vector $\in \mathbb{R}^2$ consisting of the y and z components of the vector $\mathbf{v} \in \mathbb{R}^3$
$\dot{\boldsymbol{x}}$	The time derivative of $\boldsymbol{x}$ (i.e., $d\boldsymbol{x}/dt$)
$\mathbf{v} \sim \mathcal{N}\left(\boldsymbol{\mu}, \mathbf{C}\right)$	$\mathbf{v}$ is normally distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{C}$

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 DEVCOM ARL
(PDF) FCDD RLD DCI
TECH LIB

9 APG
(PDF) FCDD RLW LE
F FRESCONI
L FAIRFAX
J BRYSON
FCDD RLW LF
J MALEY
M ILG
D EVERSON
C MILLER
B ALLIK
M HAMAOUI

1 AC PICA
(PDF) D GRASING

5 NSWCDD
(PDF) L STEELMAN
A DYKSTRA
C RIVERA
D PARKS
J LAWTON