SSC20-WKVI-08

Analysis and Design of a Sub-Optimal MEKF for Low Earth Orbit Attitude Estimation Using a Radically Inexpensive MEMS IMU

Peter J. Jorgensen, Omar F. Awad, Robert H. Bishop
University of South Florida
4202 E Fowler Ave ENG030, Tampa, FL 33620; 813-974-3780
pjorgensen@usf.edu

ABSTRACT

A previous study investigated the feasibility in using a radically inexpensive MEMS IMU with a GPS and a model of Earth's magnetic field for attitude determination. A Multiplicative Extended Kalman Filter was designed to estimate the biases and errors associated with the IMU and reduce attitude uncertainty. States with large influence on overall uncertainty were identified through error budget and sensitivity analysis. It was determined that the complexity of the Kalman filter could be significantly reduced by removing 18 of the 42-element state vector. In this study, the sub-optimal filter is designed and its feasibility for attitude determination is demonstrated through Monte Carlo simulations. The primary mission is inertial attitude control in support of spacecraft mission operations in low Earth orbit.

Our challenge is to create a radically inexpensive spacecraft using commercially available and bespoke subsystems. This inhibits the use of expensive and larger attitude sensors such as star cameras and sun sensors. While the proposed system uses a Raspberry Pi as the main flight computer, reducing computational complexity enhances the ability to provide near-real-time attitude solutions. This sub-optimal MEKF therefore improves mission capabilities by reducing computational load.

INTRODUCTION

The optimal multiplicative extended Kalman filter (MEKF) includes high-fidelity models, that include estimation parameters for biases and other errors, for all systems and sensors. While this provides a high degree of accuracy and is useful for design and simulation, the resulting filter may not be feasibly implemented on real-world hardware due to memory or computational constraints. Even when an abundance of computational power is available, reducing the complexity of the MEKF may yield faster performance, resulting in a better overall near-real-time solution and reserve compute power for other mission capabilities.

In the authors' previous work[1] it was shown that, for a spacecraft in low Earth orbit (LEO), an attitude solution of +/- 5 degrees (1-sigma) is possible using low-cost Micro Electro-Mechanical System (MEMS) Inertial Measurement Unit (IMU) and magnetometer sensors, a GPS module, and the World Magnetic Model[2] in conjunction with the TRIAD algorithm[3]. While not reasonable for missions requiring fine attitude steering (e.g. targeted high-gain RF or optical communications), this accuracy is sufficient for maximizing power generation by a sun-tracking controller for solar panel pointing. This result was enabled by a 48-state MEKF including sensor bias and error estimation for the MEMS IMU accelerometer, gyroscope, and magnetometer.

The goal of developing a sub-optimal MEKF is to enable accurate navigation within low-power systems. Several integrated guidance, navigation, and control solutions exist as commercial off the shelf systems for the small satellite market. These products must minimize resource consumption while maximizing pointing performance, which means fusing sensor data and controlling actuators with as little compute power as possible.

As part of previous work, the authors determined the error budget and sensitivity analysis for the optimal MEKF, and found that several estimated states had very little impact on overall uncertainty of position, velocity, and attitude estimate of the spacecraft[1]. Therefore, removal of these states can be done without impacting the accuracy of other state estimates and yielding a simpler MEKF. This resulting filter is the sub-optimal MEKF.

SUB-OPTIMAL MEKF

When first modeling a system for simulation and analysis, the fullest and most complete models are used to provide the best insight into system behavior. An example of this is using an atmospheric density model that accounts for time of year, solar activity, and latitude

and longitude as opposed to a model that only requires altitude. This complexity may include modeling known or assumed biases and disturbances, and even including terms for unknown or unmodeled effects, for both sensors and environmental models. The result is a more accurate model that is built from real-time environmental feedback and physics models of the system, at the cost of modeling complexity and computational load.

Suboptimal linear filter design is covered thoroughly in Gelb[4]. The concepts presented cover sensitivity analysis and error budget creation for minimizing estimation error growth for the state-reduced filter. These concepts are extended here to apply to the nonlinear MEKF.

The optimal MEKF derived previously[1] uses the following states:

$$\left.\begin{matrix}\boldsymbol{r}_{imu} \in \Re^3 \\ \boldsymbol{v}_{imu} \in \Re^3 \\ \overline{\boldsymbol{q}}_i^{\,b} \in \Re^3\end{matrix}\right\} smallsat\ parameters$$

$$\left.\begin{matrix}\boldsymbol{b}_q \in \Re^3 \\ \boldsymbol{b}_p \in \Re^3 \\ \boldsymbol{b}_v \in \Re^3\end{matrix}\right\} sensor\ biases$$

$$\left.\begin{matrix}\boldsymbol{b}_a \in \Re^3 \\ \boldsymbol{s}_a \in \Re^3 \\ \boldsymbol{\gamma}_a \in \Re^6\end{matrix}\right\} accelerometer\ bias/errors$$

$$\left.\begin{matrix}\boldsymbol{b}_g \in \Re^3 \\ \boldsymbol{s}_g \in \Re^3 \\ \boldsymbol{\gamma}_g \in \Re^6\end{matrix}\right\} gyro\ bias/errors$$

Note that the attitude state captures only the attitude deviation based on a small-angle approximation used during the quaternion rotation, and so there are three attitude state variables instead of four. An error budget and sensitivity analysis were conducted. The states related to the accelerometer errors $\{\boldsymbol{s}_a, \boldsymbol{\gamma}_a\}$ and the states related to the gyroscope errors $\{\boldsymbol{s}_g, \boldsymbol{\gamma}_g\}$ had approximately two orders of magnitude lower impact on the total estimation error budget. Hence, these states are good candidates for removal.

The sub-optimal MEKF for attitude estimation retains the system parameters position $\boldsymbol{r}_{imu}$, velocity $\boldsymbol{v}_{imu}$, and attitude $\overline{\boldsymbol{q}}_i^{\,b}$, and the position, velocity, attitude, accelerometer, and gyroscope bias terms $(\boldsymbol{b}_p, \boldsymbol{b}_v, \boldsymbol{b}_q, \boldsymbol{b}_a, \boldsymbol{b}_g)$. These terms account for the sub-optimal MEKF's 24 states. Compared to the optimal MEKF with 42 states, this is a 43% reduction in state space. The input space remains the same, with measurement vectors comprised of GPS position, GPS velocity, and attitude measurement from the TRIAD algorithm.

COMPUTATIONAL COMPLEXITY

It is difficult to accurately account for code performance using common research tools such as MATLAB, which was used for this analysis. Under the hood, these tools use optimizations that reduce runtime for common tasks, such as matrix multiplications. For illustrative purposes that may apply more to low-cost, low-power hardware on which a sub-optimal MEKF would be designed to run, a naïve accounting is presented for the MEKF update step.

Computational load can be estimated by counting the number of floating point operations required to perform a task. For example, the MEKF update step follows Equations 1-3.

$$\boldsymbol{x}_k^+ = \boldsymbol{x}_k^- + \boldsymbol{K}(\boldsymbol{y}_k - \boldsymbol{h}_k) \quad (1)$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^-(\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)^T + \boldsymbol{K}_k\boldsymbol{R}_k\boldsymbol{K}_k^T \quad (2)$$

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \quad (3)$$

Multiplying two matrices of size $\Re^{k\times l}$ and $\Re^{l\times m}$ takes $k*m*l$ multiplications and $k*m*(l-1)$ additions, and by Gaussian elimination the $m \times m$ matrix inverse takes $\frac{1}{2}m*(m+1)$ divisions, $\frac{1}{6}(2m^3+3m^2-5)$ multiplications, and $\frac{1}{6}(2m^3+3m^2-5)$ additions. Figure 1 shows the total number of floating-point operations for the MEKF state update step as a function of both state vector length n and measurement vector length m. The two markers show the computation requirement for the optimal MEKF given 42 estimated states and 9 input states and the computation requirement for the sub-optimal MEKF with 24 estimated states. Note that this complexity measurement does not account for any intermediate processing required for measurements, state propagation, or other system tasks.

Clearly, reducing the state space has significant impact on the computational complexity, which for the example of the MEKF update step is $O(n^3, m^3)$.

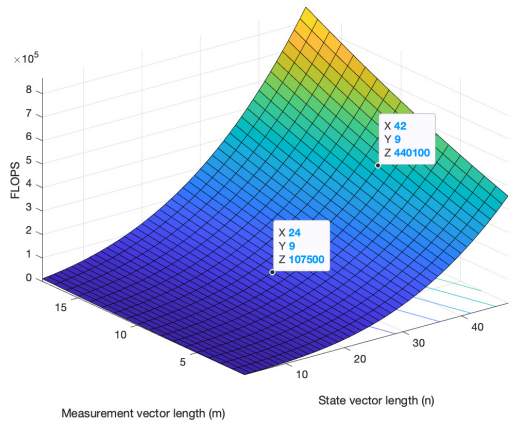

Figure 1: Computational complexity of MEKF update step

Table 1 shows a selection of common low-cost microprocessors and system on chip devices (SoCs) often used in research and industry for embedded electronics control and data processing. Million instructions per second (MIPS) is used to compare computing performance. Note that this does not guarantee performance for MEKF applications, since some hardware (e.g. the ATmega328) is not well-suited for floating point math or programs requiring large amounts of memory. One device (Cortex-M4F) specifically includes a floating-point unit (FPU) that is useful for MEKF type calculations.

Table 1: Microcontroller and SoC performance comparison

MCU	CPU Type	MIPS	Clock (MHz)
ATmega328	8-bit	20	20
MSP430x6xx	16-bit	25	25
ARM Cortex-M3	32-bit	45	50
ARM Cortex-M4F	32-bit w/ FPU	45	50
ARM Cortex-A8	32-bit	2,000	1,000
ARM Cortex-A53	64-bit	2,688	1,200
ARM Cortex-A9	32-bit	13,800	1,500

The complexity analysis shows that the optimal MEKF update step requires 0.4M instructions, and for the reduced state space sub-optimal MEKF the update step requires 0.1M instructions. Standard MEKF processing includes state propagation and measurement processing, which are assumed to be of similar complexity to the update (but can occur at different rates). Table 2 shows the total MIPS required for each of the optimal and sub-optimal MEKFs presented at the given rates.

Table 2: MEKF estimated processing requirements

Step	Instructions (M)	Rate (Hz)	MIPS
	Optimal		
Update	0.4	5	2.0
Propagate	0.4	50	20.0
Measurements	0.2	15	3.0
			25.0
	Sub-optimal		
Update	0.1	5	0.5
Propagate	0.1	50	5.0
Measurements	0.05	15	0.75
			6.25

Running the optimal MEKF on low-cost microcontrollers such as the MSP430 is not feasible in real-time due to raw instructions per second processing requirements. The optimal MEKF might run in real time on the ARM Cortex-M3 and Cortex-M4F. The estimates in Table 2 show a 75% reduction in estimated instructions per second of the sub-optimal MEKF over the optimal MEKF. Based on this estimate of required compute power, it could be feasible to use the ultra-low power MSP430 to run the sub-optimal filter, or to use the more powerful processors in a reduced-power state.

It is important to note that the word size of instructions and data processing capabilities was not taken into account when comparing the processors. It may be feasible on an instructions-per-second metric to run the sub-optimal MEKF on the MSP430, but it may not yield useful results due to the 16-bit nature of the device. Comparing N-bit floating point accuracy is outside the scope of this work.

The lowest-power device, the ATmega328, offered sufficient compute performance to implement the sub-optimal MEKF, although other factors, mainly the 8-bit word size, make it a poor choice for the application. The

other low-power processors could support an embedded MEKF, although the optimal filter would likely stress the limits of these low power devices. The ARM Cortex-A8, Cortex-A53, and Cortex-A9 are included for comparison because these are commonly used in single-board computers often used for hobbyist and educational projects. They are all overpowered for the example problem of running an embedded MEKF. However, if the target spacecraft is to only have one computer, these are good choices due to the large amount of processing power yielding sufficient margin for other system requirements. The single-board computers are useful in the design and prototyping phases of system development as they often include general purpose input/output (GPIO) headers for connecting peripheral devices. Any of these choices would handle the workload from the optimal MEKF, although it is still of interest to optimize the filter and implement the sub-optimal MEKF to reduce power consumption and thermal waste energy generation.

SIMULATION AND RESULTS

For low Earth orbit navigation, typical spacecraft are size, weight, and power constrained. Selecting a low-power CPU has system-wide benefits in terms of power and thermal management. It is also interesting to consider a self-contained guidance, navigation, and control subsystem that fuses sensors and actuators as an off-the-shelf solution. Both scenarios require careful balancing of system resources, and minimizing the MEKF state space is done to reduce computation while maintaining state estimate accuracy.

The optimal MEKF from the authors' previous work was reduced by 18 state variables into the sub-optimal MEKF for this study. Figure 2 through Figure 10 show Monte-Carlo simulation results for position, velocity, and attitude error of the optimal and sub-optimal MEKFs. The Monte-Carlo results capture 500 individual simulation runs for both the optimal and sub-optimal MEKFs using the same random number generator seed to ensure equal comparison. The following figures show averages created from all Monte-Carlo runs. MATLAB R2019b Update 2 was used for the simulations.

Figures Figure 11 through Figure 13 compare the error covariance for position, velocity, and attitude, showing the percent difference in error covariance estimate of the sub-optimal MEKF compared to the optimal MEKF.

As can be seen, the sub-optimal MEKF performs within about 10% of the optimal MEKF, while the reduction in computation is about 75%.

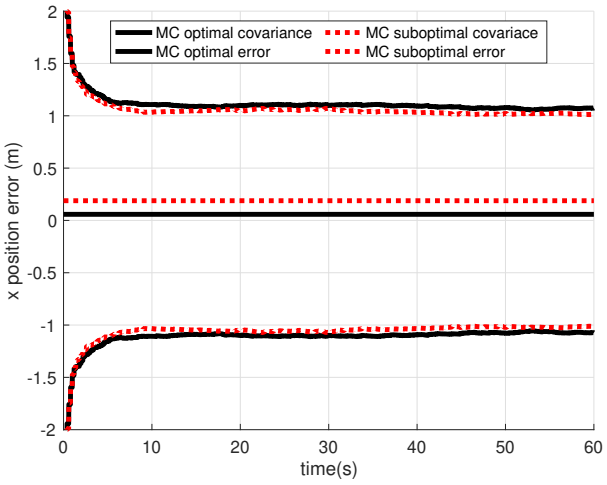

Figure 2: X position error and covariance comparison

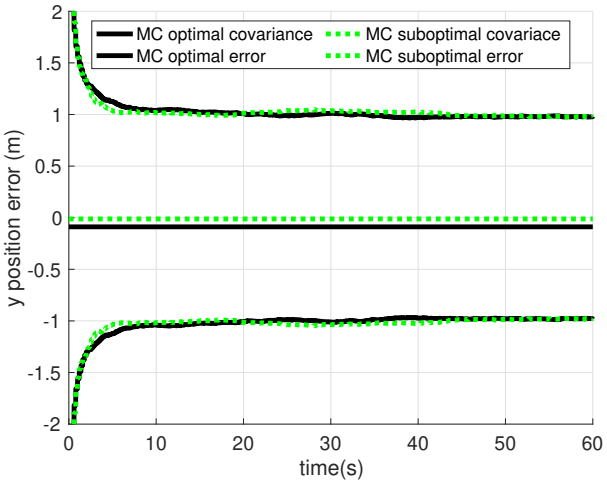

Figure 3: Y position error and covariance comparison

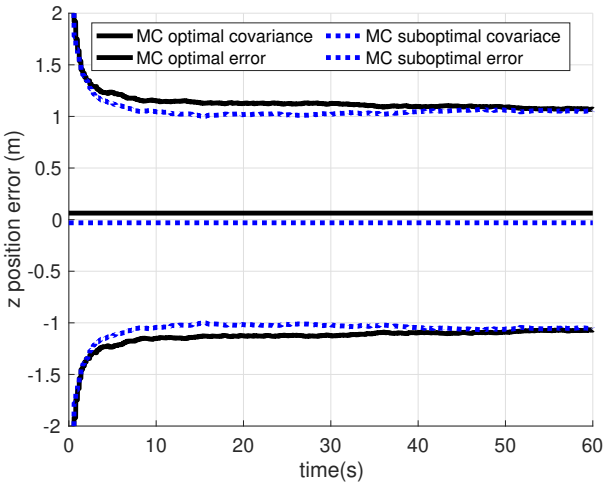

Figure 4: Z position error and covariance comparison

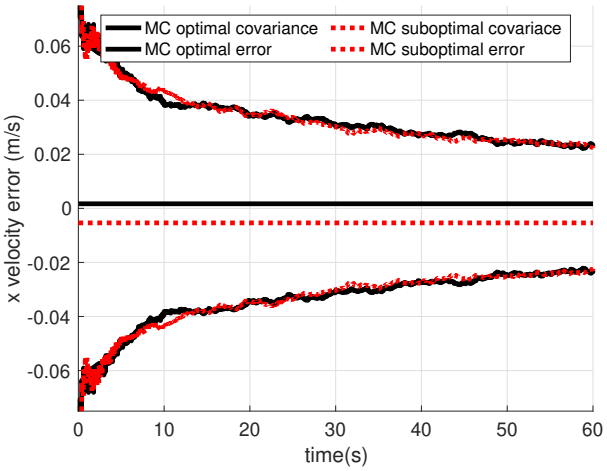

Figure 5: X velocity error and covariance comparison

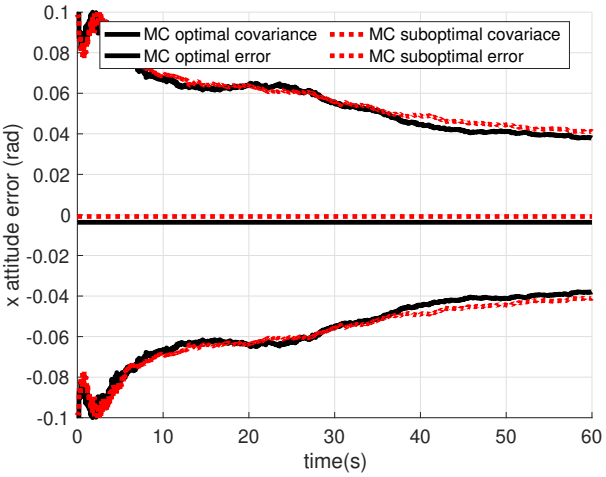

Figure 8: X attitude error and covariance comparison

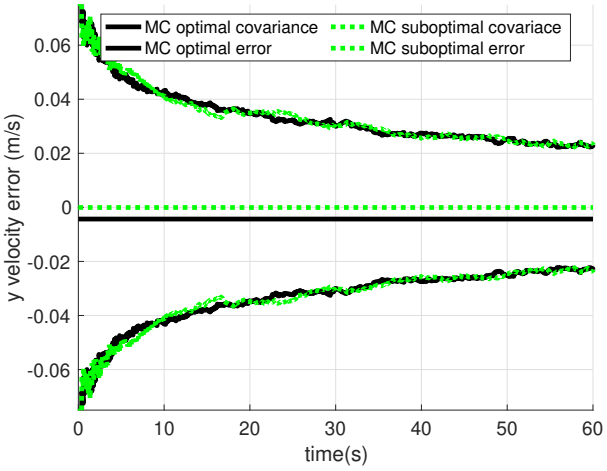

Figure 6: Y velocity error and covariance comparison

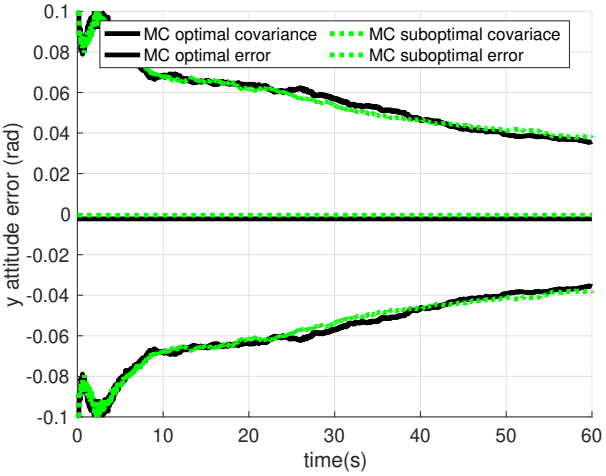

Figure 9: Y attitude error and covariance comparison

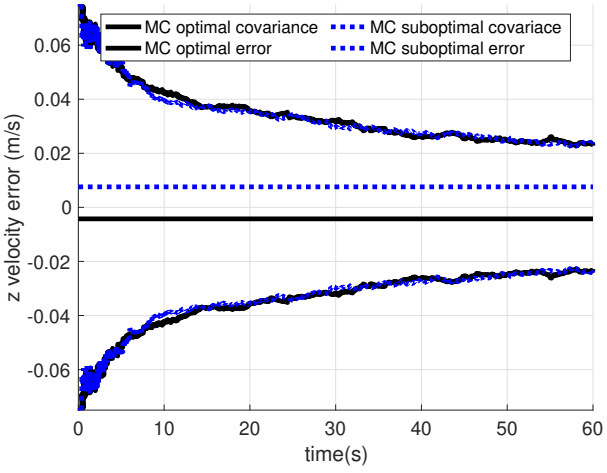

Figure 7: Z velocity error and covariance comparison

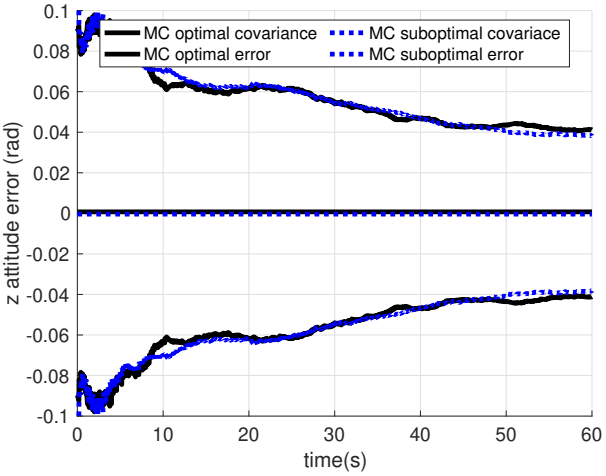

Figure 10: Z attitude error and covariance comparison

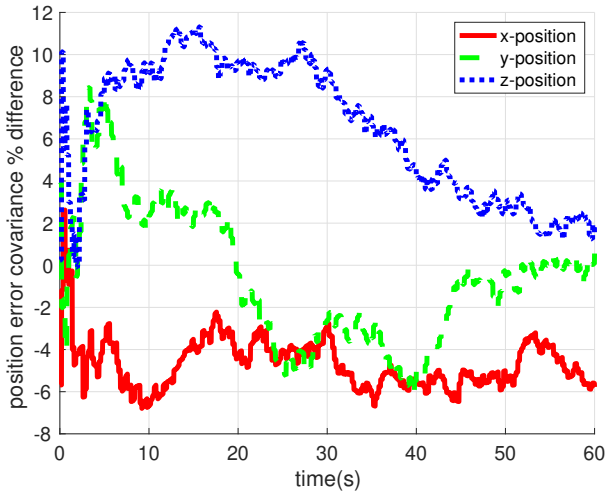

Figure 11: Position error covariance sub-optimal vs. optimal filter percent difference

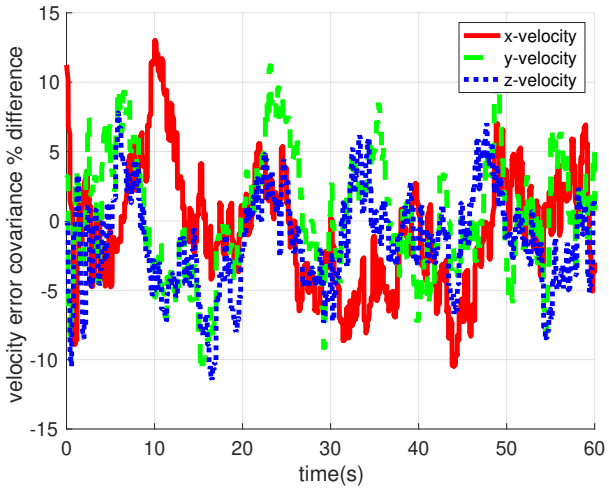

Figure 12: Velocity error covariance sub-optimal vs. optimal filter percent difference

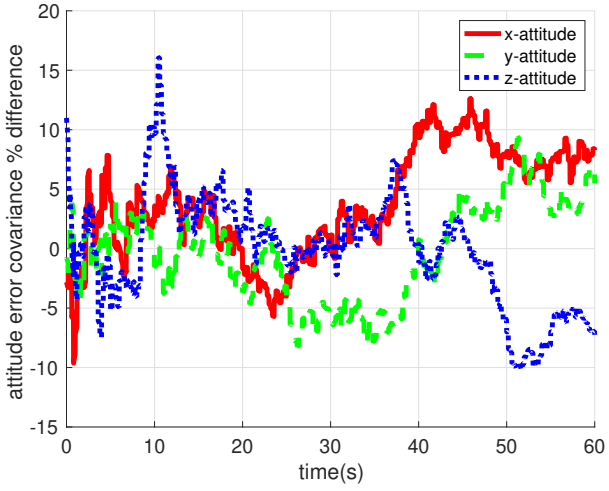

Figure 13: Attitude error covariance sub-optimal vs. optimal filter percent difference

CONCLUSION

The optimal MEKF system from authors' prior work was compared to a sub-optimal MEKF, where 18 state variables were removed based on sensitivity and error budget analyses. An estimated computational resource requirement analysis was presented, and several common low-power and higher-power processors compared. The low power options (MSP430, Cortex-M3, Cortex-M4F) were deemed feasible for use as processors for an embedded MEKF, and the higher power options (Cortex-A8, Cortex-A53, Cortex-A9) were deemed feasible for supporting both the MEKF and other system processing requirements.

Finally, the sub-optimal MEKF was compared to the optimal MEKF in position, velocity, and attitude estimation error. Results showed the state reduction of the sub-optimal filter succeeded in maintaining overall estimation error bounds: the error variance values were within about 10% of the optimal filter values. This minimal additional error came with a computational reduction of about 75%. For low power or embedded applications, this was deemed a reasonable trade-off.

Extending this work might include investigating the observability of the removed states, since unobservable state variables might indicate incorrectly modeled or coupled effects. Also, hardware-specific characterization of low-cost sensors, like the simulated MEMS IMU, to measure actual noise and disturbance bounds, can be done to inform further development of realistic models.

References

1. Awad, O.F. and R.H. Bishop, "A Multiplicative Extended Kalman Filter for Low Earth Attitude Estimation Using a Radically Inexpensive MEMs IMU in a 0.5U Cubesat," 43rd Annual AAS Guidance, Navigation & Control Conference, Colorado Springs, CO, January 2020.
2. Chulliat, A., S. Macmillan, et al., "The US/UK World Magnetic Model for 2015-2020," BGS and NOAA, 2015.
3. Bar-Itzhack, I.Y. and R.R. Harman, "Optimized TRIAD Algorithm for Attitude Determination," Journal of Guidance, Control, and Dynamics, vol. 20, No. 1, January 1997.
4. Gelb, A., "Applied Optimal Estimation," MIT Press, 1974.