Implementation of a Modular Linear and Unscented Kalman Filter

Will MacCormack [1] Benjamin Rubinstein [1]

[1]Case Western Reserve University Department of Aerospace Engineering

Abstract

- We present an implementation of a modular Kalman filter, utilizing both linear and Unscented Kalman filtering elements, optimized for robust localization in aerospace and robotics applications.
- The architecture utilizes modular predictor and corrector components, with flexibility for system-specific evolution and observation functions, ensuring adaptability.
- Two synthetic modalities were tested and residual values were found to be within acceptable ranges
- Proposed enhancements include transitioning from Python to a compiled language and treating select parameters as time-independent Gaussian random variables.

Introduction

- State estimation is a critical part of many control algorithms for applications in robotics, aerospace engineering, and mechanical systems.
- The simplicity and Gaussian nature of Newtonian systems has led to a proliferation of Kalman Filters, which exploit the fact that a Gaussian distribution is fully described by its mean and covariance.
- The Linear Kalman Filter directly propagates mean and covariance through a linear evolution model and computes a posteriori estimates and covariance by solving the linear inverse problem. This reliance on a linear evolution and sensor model greatly reduces the set of applicable systems.
- The Extended Kalman filter uses the Jacobian of the system to linearize the evolution and sensor models.
- The UKF models the probability distribution using an ensemble of 'sigma points' which are used for both predictive and corrective steps.
- All Bayesian filters rely upon evolutionary and observational models to propagate state estimates and incorporate observations, respectively. The simplified evolution and observation models will be discussed in applications and on the specific equation derivations.

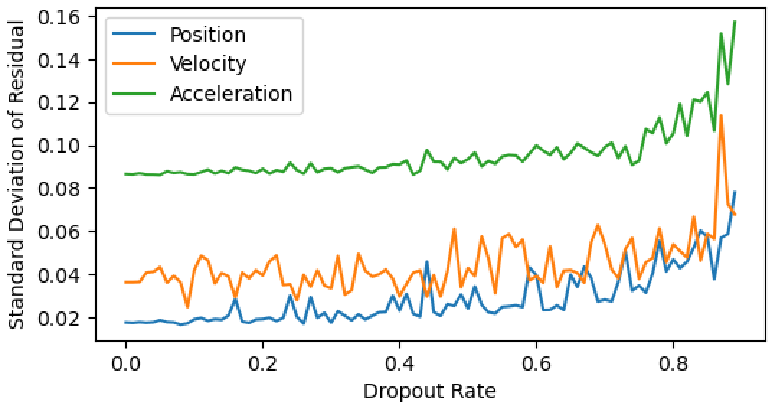
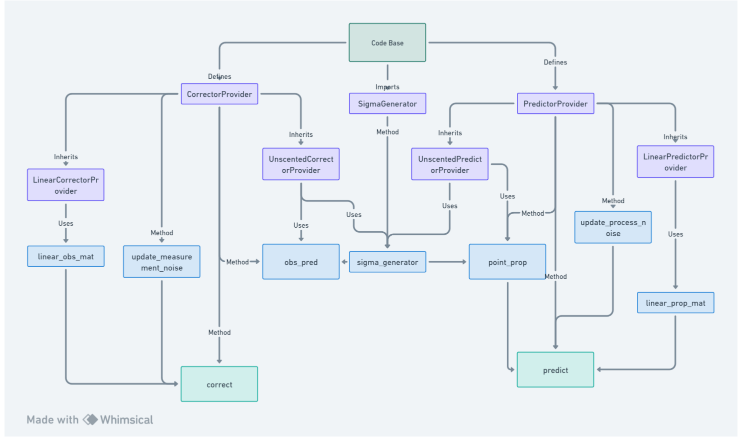


Figure 1. Residuals for different rates of sensor dropout



Mathematical Model and Update Equations

Variables

x is the state, b is the observation, D is the covariance. K is the Kalman Gain Matrix. Γ and Σ are the process and measurement noise estimates.

Linear Kalman Filter Update

Linear Predictor: $\hat{\bar{x}} = F\bar{x}, \quad \hat{D} = FDF^T + \Gamma$

Linear Corrector: $\Delta = b - Ax, \quad K = DA^T(ADA^T + \Sigma)^{-1}$

Linear Update: $\bar{x}' = \bar{x} + K\Delta, \quad D' = (I - KA)D$

Unscented Kalman Filter Update

Sigma Points and Weights:

$x^{(0)} = \bar{x} \quad x^{(i)} = \bar{x} + (\alpha\sqrt{\kappa D})_i \quad \forall i = 1, \ldots, n$

$x^{(j)} = \bar{x} - (\alpha\sqrt{\kappa D})_{j-n} \quad \forall j = n+1, \ldots, 2n$

$w_m^{(0)} = \frac{\alpha^2\kappa - n}{\alpha^2\kappa}, \quad w_c^{(0)} = w_m^{(0)} - \alpha^2 + \beta$

$w_m^{(i)} = w_c^{(i)} = \frac{1}{2\alpha^2\kappa} \quad \forall i = 1, \ldots, 2n$

Unscented Prediction: $\hat{x}_{(i)} = f(x^{(i)}, \Delta t)$, $\hat{\bar{x}} = \sum_{i=0}^{2n} w_m^{(i)}\hat{x}^{(i)}$, $\hat{D} = \sum_{i=0}^{2n} w_c^{(i)}(x^{(i)} - \hat{\bar{x}})(x^{(i)} - \hat{\bar{x}})^T + \Gamma$

Correction: $z^{(i)} = h(x^{(i)}), \quad \hat{z} = \sum_{i=0}^{2n} w_m^{(i)} z^{(i)}$

Measurement Predicted Covariance: $S = \sum_{i=0}^{2n} w_c^{(i)}(z^{(i)} - \hat{z})(z^{(i)} - \hat{z})^T + \Sigma$

Cross Covariance: $T = \sum_{i=0}^{2n} w_c^{(i)}(x^{(i)} - \hat{x})(z^{(i)} - \hat{z})^T$

Kalman Gain: $K = TS^{-1}$

Update: $\bar{x} = \hat{\bar{x}} + K(b - \hat{z}), \quad D = \hat{D} - KSK^T$

Implementation Strategy

- Our modular Kalman filter implementation separates filtering into predictors and correctors. Predictors evolve the state estimate through time, while correctors handle a single, atomic observation type.
- Inheritance and Polymorphism were used to implement inter-operable Linear, Extended, and Unscented Kalman Filter update steps with few lines of code. These classes only provided mean and covariance update functions, abstracting evolution and observation models to user-specific implementation.
- A single predictor evolves the system each timestep, and each independent sensor update has its own corrector to handle different update rates between sensors and enable model function despite data delays or dropouts.

Results

- We present state vs. time and residual vs. time graphs for the two tested modalities.
- The model was first tested on a spring-mass oscillator. The filter tracked one degree of position, velocity, and acceleration without knowledge of the underlying spring-mass system.
- Second, the filter was applied to a modeled rocket flight, where the state included three axis position, velocity, acceleration, and rotational velocity along with a quaternion-based orientation.
- Dropout rates of 10% through 99% were tested on the spring-mass system, and model predictive quality remained mostly unaffected up to sensor dropout of 80%.
- Residuals for position and orientation of the rocket were found to be within acceptable ranges (<15%) during ascent, proving on simulated data that the model could correct without knowledge of the rocket's mass or thrust.

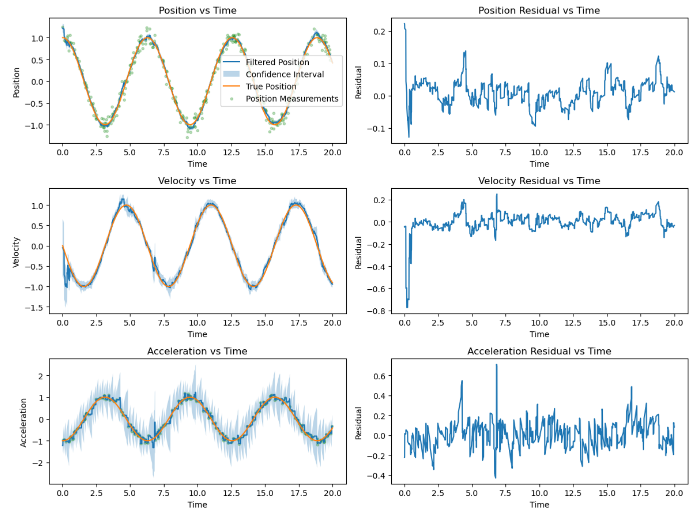


Figure 2. Linear Kalman Filter, Dropout 50%

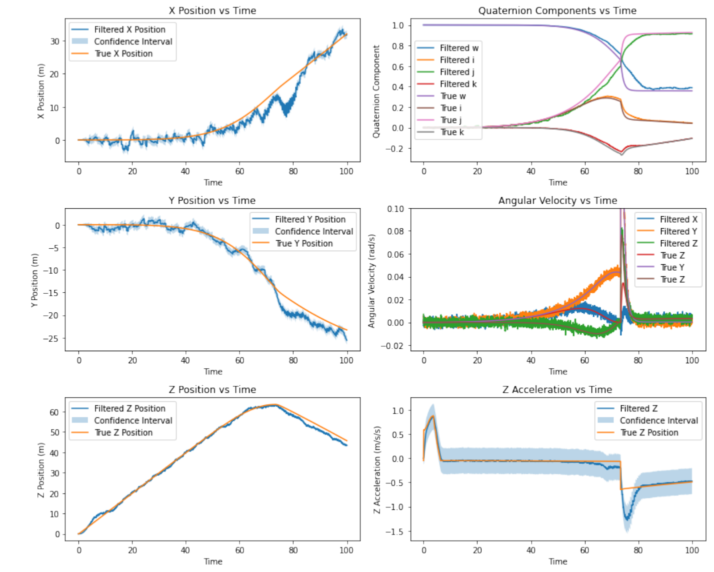


Figure 3. Rocket Filter Application

Future Work

- The filter applicability could be improved by transitioning from Python to a compiled language like C++ or Rust, accelerating performance and enabling implementation on embedded hardware for real-time, in situ localization.
- Integration efforts could be reduced by automatically approximating covariances.

Conclusions

- Through this work, we have shown the effectiveness of a modular Kalman filter for sensor fusion localization and state estimation.
- By creating an abstract base system and adding functionality for each specific filter and system, the flexibility enables the same code to be used for a variety of projects.
- Future work will involve compiling the code for embedded deployments and applying the model real data.

References

[1] James Maley. A modular approach to kalman filter design and analysis. Technical report, DEVCOM Army Research Laboratory, March 2021. Pagination: 60 pages.

[2] F. Landis Markley. Attitude estimation or quaternion estimation? In *Flight Mechanics Symposium 2003*, page N/A, Greenbelt, MD, USA, 2003. National Aeronautics and Space Administration. NASA/CP-2003-212245.

[3] Eric A. Wan and Rudolph van der Merwe. The unscented kalman filter for nonlinear estimation. Technical report, Oregon Graduate Institute of Science & Technology, 2000.