Kalman Filter for Missile State Estimation

Alonzo Lopez

March 10, 2021

1 Abstract

A continuous-time Kalman Filter is implemented to estimate the relative states in a missile intercept operation. The filter's performance is first verified via Monte Carlo simulation of a Gauss-Markov process driven by a random forcing function with an exponential correlation. The filter's robustness is then confirmed in a second Monte Carlo simulation where the dynamic model is driven by a random telegraph signal instead of the random forcing function.

2 Introduction

The missile intercept problem illustrated in Figure 1 features a pursuer attempting to close the distance between itself and a maneuvering target. This problem has two parts: estimation and control. The Kalman Filter detailed in this paper solves the estimation problem given a stochastic process model and sensor noise model.

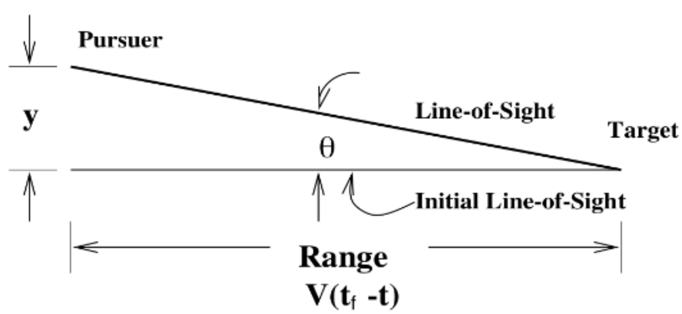

Figure 1: Missile Intercept Illustration.

3 The Dynamic Model

The dynamics of the problem are

$$\begin{aligned} \dot{y} &= v \\ \dot{v} &= a_P - a_T \end{aligned} \tag{1}$$

where a_P, the acceleration of the pursuer, is known to be zero. The input, a_T, is the target acceleration and is treated as a random forcing function with an exponential correlation,

$$\begin{aligned} E[a_T] &= 0 \\ E[a_T(t)a_T(s)] &= E[a_T^2]e^{\frac{-|t-s|}{\tau}} \end{aligned} \tag{2}$$

The scalar, τ, is the correlation time. The initial lateral position, $y(t_0)$, is zero by definition. The initial lateral velocity, $v(t_0)$, is random and assumed to be the result of launching error:

$$\begin{array}{llllll} E[y(t_0)] & = 0 & E[v(t_0)] & = 0 & & \\ E[y(t_0)^2] & = 0 & E[y(t_0)v(t_0)] & = 0 & E[v(t_0)^2] & = \text{given} \end{array}$$

The measurement, z, consists of a line-of-sight angle, θ. For $|\theta| << 1$,

$$\theta \approx \frac{y}{V_c(t_f - t)} \tag{3}$$

It will also be assumed that z is corrupted by fading and scintiallation noise so that

$$\begin{aligned} z &= \theta + n \\ E[n(t)] &= 0 \\ E[n(t)n(\tau)] &= V\delta(t-\tau) = \left[R_1 + \tfrac{R_2}{(t_f-t)^2}\delta(t-\tau)\right] \end{aligned} \tag{4}$$

The process noise spectral density, W, is

$$W = GE[a_T^2]G^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & E[a_T^2] \end{bmatrix} \tag{5}$$

Given the above, the state-space equations for the process and measurement are

$$\begin{bmatrix} \dot{y} \\ \dot{v} \\ \dot{a}_T \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -\frac{1}{\tau} \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} y \\ v \\ a_T \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}}_{B} a_P + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}_{G} w_{a_T} \tag{6}$$

$$z = \underbrace{\begin{bmatrix} \frac{1}{Vc(t_f-t)} & 0 & 0 \end{bmatrix}}_{H} \begin{bmatrix} y \\ v \\ a_T \end{bmatrix} + n \tag{7}$$

The following values are used in practice to simulate the model.

$$V_c = 300\frac{ft}{sec} \qquad E[a_T^2] = (100\frac{ft}{sec^2})^2 \quad t_f = 10sec \quad R_1 = 15\times10^{-6}\frac{rad^2}{sec}$$
$$R_2 = 1.67\times10^{-3}\frac{rad^2}{sec^3} \qquad \tau = 2sec \qquad b = 1.52\times10^{-2}$$

4 The Kalman Filter Algorithm

The continuous-time Kalman Filter has the form

$$\begin{aligned}\dot{\hat{y}} &= \hat{v} + K_1 \underbrace{\left(z - \tfrac{\hat{y}}{V_c(t_f-t)}\right)}_{residual} \\ \dot{\hat{v}} &= -\hat{a}_T + K_2\left(z - \tfrac{\hat{y}}{V_c(t_f-t)}\right) \\ \dot{\hat{a}}_T &= -\frac{\hat{a}_T}{\tau} + K_3\left(z - \tfrac{\hat{y}}{V_c(t_f-t)}\right)\end{aligned} \tag{8}$$

Where the gains are

$$\begin{aligned}K_1 &= \frac{p_{11}}{V_c R_1(t_f-t) + \frac{V_c R_2}{t_f-t}} \\ K_2 &= \frac{p_{12}}{V_c R_1(t_f-t) + \frac{V_c R_2}{t_f-t}} \\ K_3 &= \frac{p_{13}}{V_c R_1(t_f-t) + \frac{V_c R_2}{t_f-t}}\end{aligned} \tag{9}$$

The scalars, p_{ij},are the (i, j) elements of the error covariance matrix that is propagated by the Ricatti equation,

$$\dot{P} = FP + PF^T - \frac{1}{V_c^2 R_1 (t_f-t)^2 + V_c^2 R_2} P\bar{H}^T\bar{H}P + W \tag{10}$$

where $\bar{H} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}$.

5 Simulation Results and Discussion

5.1 Simulation Initialization

Every simulation run has its true states, y, v, and a_T, initialized by drawing from a zero-mean Gaussian with covariance values as indicated along the diagonal of the below initial covariance

$$P(0) = \begin{bmatrix}\underbrace{0}_{E[y(t_0)^2]} & 0 & 0 \\ 0 & \underbrace{(200\frac{ft}{sec})^2)}_{E[v(t_0)^2]} & 0 \\ 0 & 0 & \underbrace{(100\frac{ft}{sec^2})^2)}_{E[a_T^2]}\end{bmatrix} \tag{11}$$

The true states are propagated via Euler integration using (6), and a noise sequence is generated to provide the simulation values of w_{a_T} at every time instant by drawing from a Gaussian with statistics $w_{a_T}(k) \sim \mathcal{N}(0, E[a_T^2])$. For each discrete-time instant in the simulation, the sensor measurement is simulated using (3) and (4).

5.2 Analysis of One Realization

The Kalman filter is run offline with gains and covariance computed a priori. The Kalman gains are illustrated in Figure 2 and the a priori covariance values are plotted in Figure 3. Both the Kalman gains and a priori covariance plots match the expected plots.

The filter's performance is then tested and plotted for one realization of the Gauss-Markov process. Figures 4, 5, 6 plot both the true and estimated relative position, relative velocity, and target acceleration, respectively. The filter is shown to track all three states well for this realization.

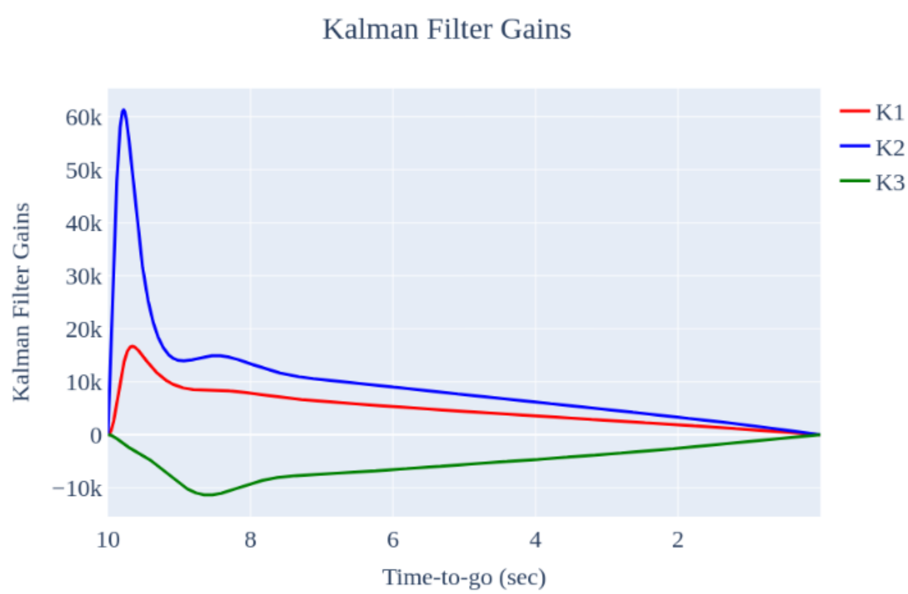

Figure 2: The Kalman gains computed a priori.

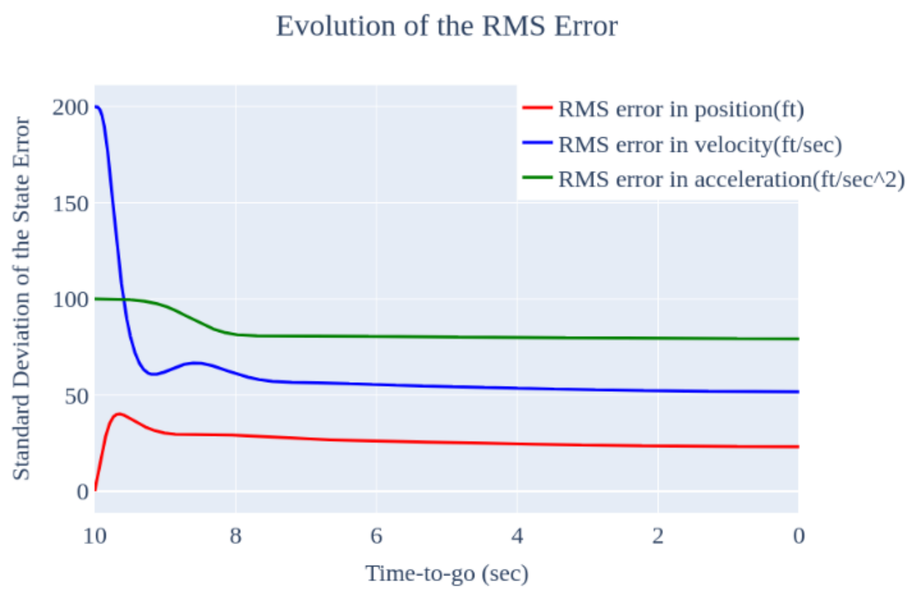

Figure 3: The evolution of the RMS error computed a priori.

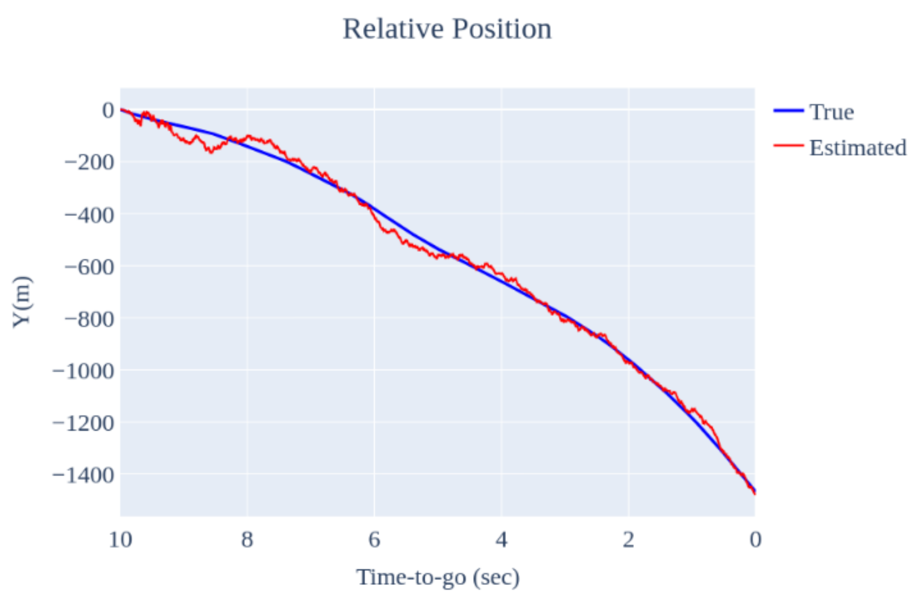

Figure 4: The true and estimated relative distance along y between the pursuer and target in one realization of the Gauss-Markov model.

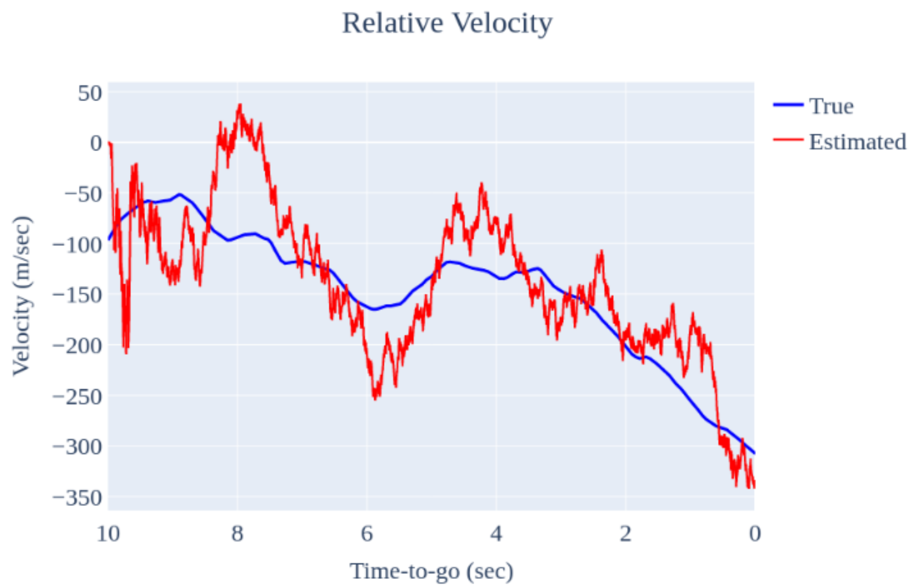

Figure 5: The true and estimated relative velocity along y between the pursuer and target in one realization of the Gauss-Markov model.

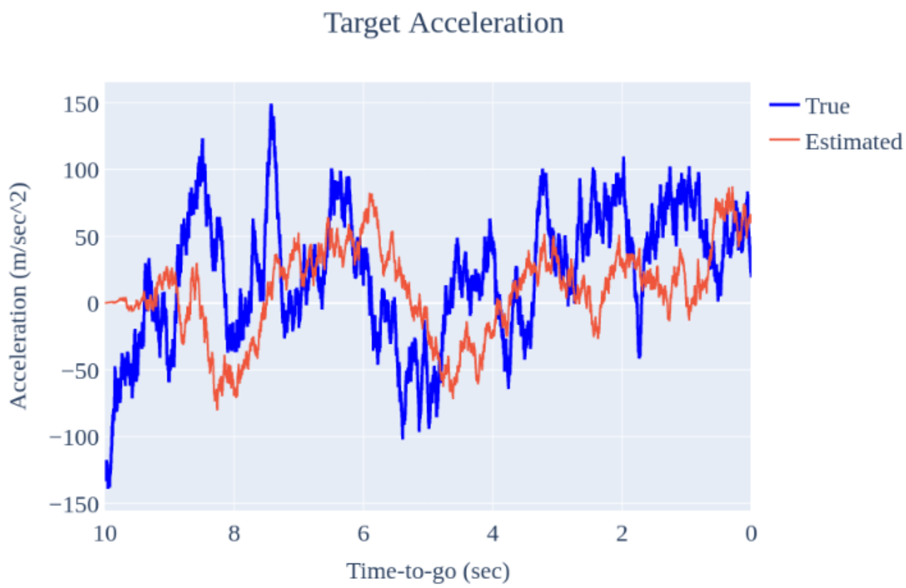

Figure 6: The true and estimated target acceleration in one realization of the Gauss-Markov model.

5.3 Analysis of the Error Variance

To compare the actual performance of the filter to the precomputed covariance values, the Root Mean Square Error (RMSE) in the filter's state estimate is calculated for a Monte Carlo simulation with 10,000 realizations. Figures 7, 8, and 9 plot the actual and a priori RMSE in position, velocity, and acceleration, respectively. The figures show that the actual RMSE plots in the three states match the precomputed values, thus verifying the filter's performance for the Gauss-Markov model.

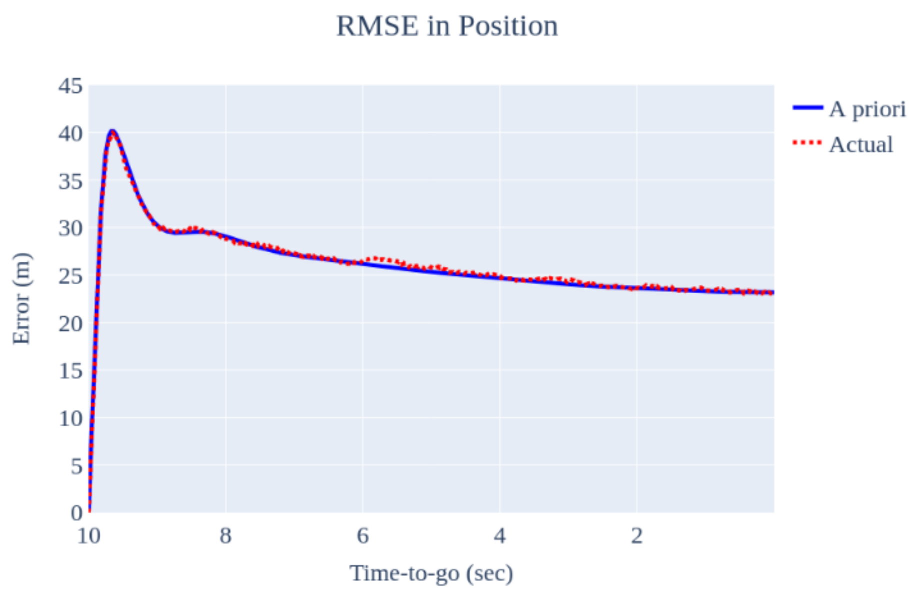

Figure 7: RMS error in position.

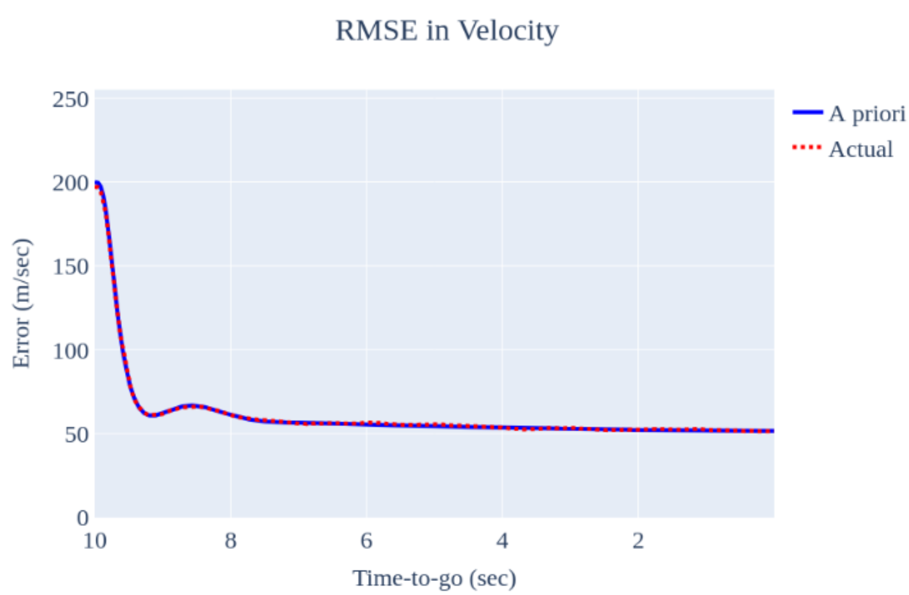

Figure 8: RMS error in velocity.

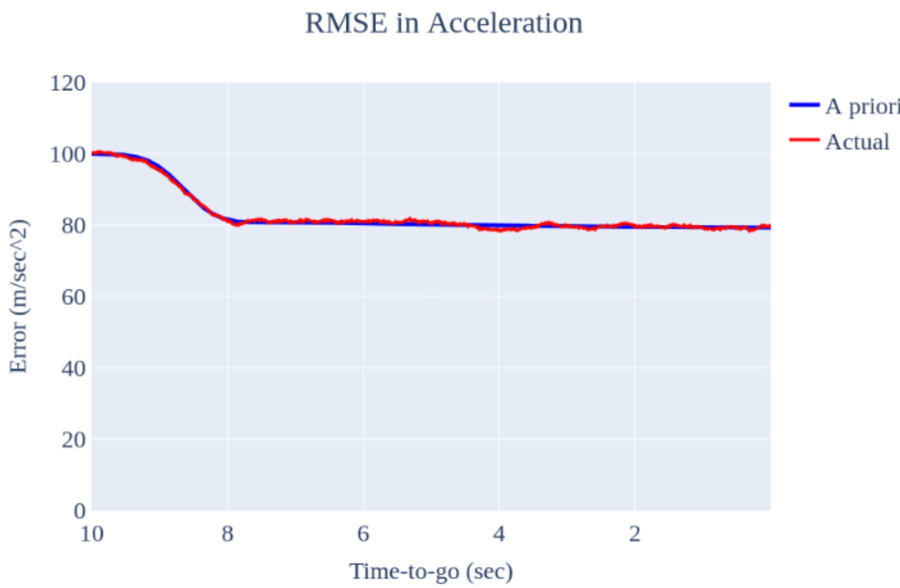

Figure 9: RMS error in acceleration.

The residual process is also shown to be uncorrelated in time at two time points throughout the Monte Carlo simulation. For $m = 5000$ and $q = 7500$, the ensemble average for the correlation of the residuals is

$$\frac{1}{N_{ave}} \sum_{l=1}^{N_{ave}} r^l(t_m) r^l(t_q)^T = 0.0009398 \approx 0 \tag{12}$$

6 Filter Robustness

6.1 The Telegraph Model

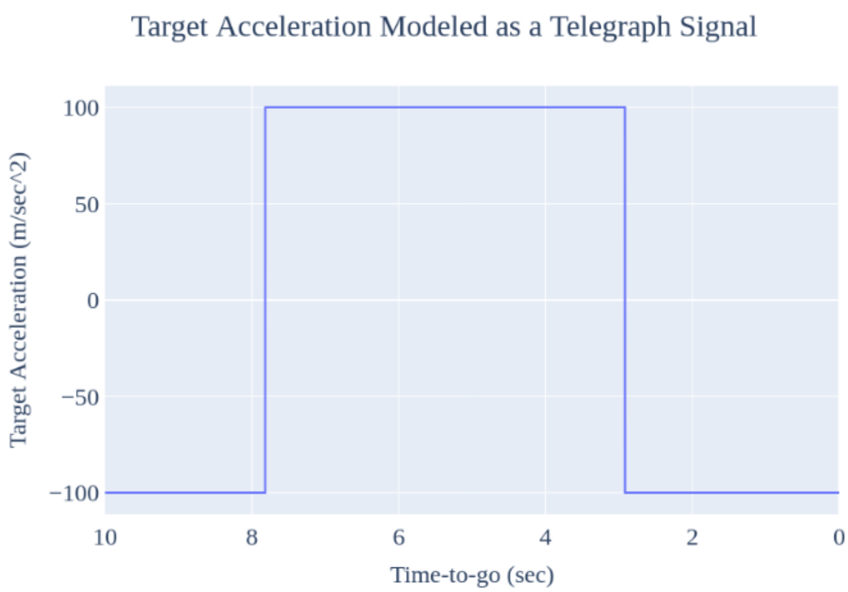

Figure 10: A close-up view of the target acceleration as modeled by the randomly switching telegraph signal.

To characterize the robustness of the filter initially designed for the Gauss-Markov process detailed by (6) and (7), we replace the process dynamics of (6) with the below dynamics

$$\begin{bmatrix} \dot{y} \\ \dot{v} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} y \\ v \end{bmatrix}}_{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} a_T \tag{13}$$

where a_T is now modeled by a random telegraph signal that maintains magnitude $a_T = 100\frac{ft}{sec^2}$ but switches sign randomly with Poisson probability. A realization of this signal is illustrated in Figure 10. The telegraph signal is initalized with value $a_T(0) = \pm a_T$ according to the following probabilities: $P(a_T(0) = a_T) = 0.5$ and $P(a_T(0) = -a_T) = 0.5$. The probability of k sign switches in a time interval of length T is $P(k(T))$.

$$P(k(T)) = \frac{(\lambda T)^k e^{-\lambda T}}{k!} \tag{14}$$

We can use (14) to generate switching times for the signal. Let $T = t_{n+1} - t_n$ be the time between two switch times. The probability that no switch occured in T is

$$P(T' > T|t = t_n) = P(\text{zero sign changes in T}) = e^{-\lambda T} \tag{15}$$

Conversely, the probability that at least one change occured is

$$P(T' \leq T|t = t_n) = 1 - e^{-\lambda T} \tag{16}$$

To produce the random switch time, t_{n+1}, draw the random variable U from a uniform density function spanning the range [0,1] then set that equal to (16) and solve for t_{n+1}.

$$\begin{aligned}
1 - e^{-\lambda T} &= U \\
e^{-\lambda T} &= 1 - U \\
-\lambda T &= ln(1 - U) \\
T &= \frac{-1}{\lambda} ln(1 - U) \\
t_{n+1} &= t_n - \frac{1}{\lambda} ln(1 - U)
\end{aligned}$$

Since $1 - U$ is also a uniform density function, then

$$t_{n+1} = t_n - \frac{1}{\lambda} ln(U) \tag{17}$$

(17) is used to generate the switching times in the signal illustrated in Figure 10.

The statistics of the telegraph signal are such that the mean and autocorrelation match that of the Gauss-Markov process. To show that the telegraph signal's mean is zero, we start by finding the probability of an even number of sign changes in T

$$P(k(T)\ is\ even) = \sum_{k=0, k\ even}^{\infty} \frac{(\lambda T)^k e^{-\lambda T}}{k!} = e^{-\lambda T} \sum_{k=0}^{\infty} \frac{(1 + (-1)^k)(\lambda T)^k}{2k!} \tag{18}$$

Since $e^{\lambda T} = \sum_{k=0}^{\infty} \frac{(\lambda T)^k}{k!}$, then

$$\begin{aligned}
P(k(T)\ is\ even) &= e^{-\lambda T} \left[\sum_{k=0}^{\infty} \frac{(\lambda T)^k}{2k!} + \frac{(-\lambda T)^k}{2k!} \right] \\
P(k(T)\ is\ even) &= \frac{1}{2} e^{-\lambda T} \left[e^{\lambda T} + e^{-\lambda T} \right] \\
P(k(T)\ is\ even) &= \frac{1}{2} \left[1 + e^{-2\lambda T} \right]
\end{aligned} \tag{19}$$

By a similar process, the probability that there are an odd number of sign changes in T is

$$P(k(T)\ is\ odd) = \frac{1}{2} \left[1 - e^{-2\lambda T} \right] \tag{20}$$

The the probability that $a_T(t)$ is positive is

$$\begin{aligned} P(a_t(t) = a_T) &= P(a_T(t) = a_T | a_T(0) = a_T)P(a_T(0) = a_T) \\ &+ P(a_T(t) = a_T | a_T(0) = -a_T)P(a_T(0) = -a_T) \\ &= \frac{1}{2}P(\text{k is even for T=[0,t]}) + \frac{1}{2}P(\text{k is odd for T=[0,t]}) \\ &= \frac{1}{2}\left\{\tfrac{1}{2}\left[1 + e^{-2\lambda t}\right] + \tfrac{1}{2}\left[1 - e^{-2\lambda t}\right]\right\} = \frac{1}{2} \end{aligned}$$

Thus, the mean acceleration is

$$\bar{a}_T = a_T P(a_T(t) = a_T) - a_T P(a_T(t) = -a_T) = 0$$

To set λ so the telegraph signal's autocorrelation matches that of the Gauss-Markov model, we first formulate the autocorrelation as

$$\begin{aligned} R_{a_T a_T}(t_1, t_2) &= E[a_T(t_1)a_T(t_2)] \\ &= a_T^2 P(a_T(t_2) = a_T(t_1)) - a_T^2 P(a_T(t_2) \neq a_T(t_1)) \\ &= a_T^2 \frac{1}{2}\left[1 + e^{-2\lambda|t_2 - t_1|}\right] - a_T^2 \frac{1}{2}\left[1 - e^{-2\lambda|t_2 - t_1|}\right] = a_T^2 e^{-2\lambda|t_2 - t_1|} \end{aligned}$$

If $\frac{1}{\tau} = 2\lambda$, then the autocorrelation of the Gauss-Markov process is the same as the random telegraph signal $(t_2 - t_1) = (t - s)$, and the means of both are zero. To make that the case for $\tau = 2$, we set $\lambda = 0.25$.

6.2 Analysis of One Realization

To test the performance of the filter on the Telegraph model, a single realization is simulated and plotted. Figure 11 plots the true and estimated relative position, and Figure 12 plots the true and estimated relative velocity. The performance of the filter is comparable to its performance with the Gauss-Markov model.

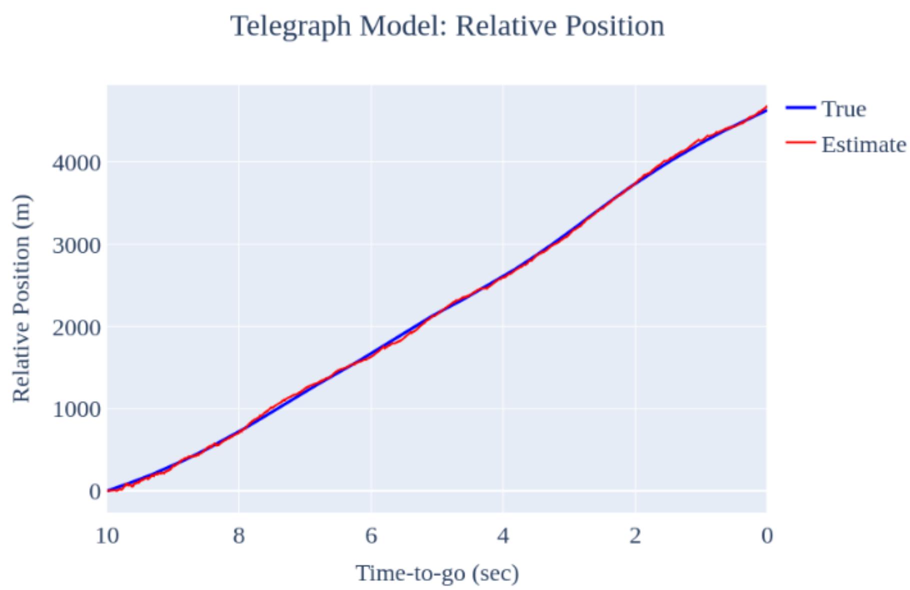

Figure 11: Telegraph position

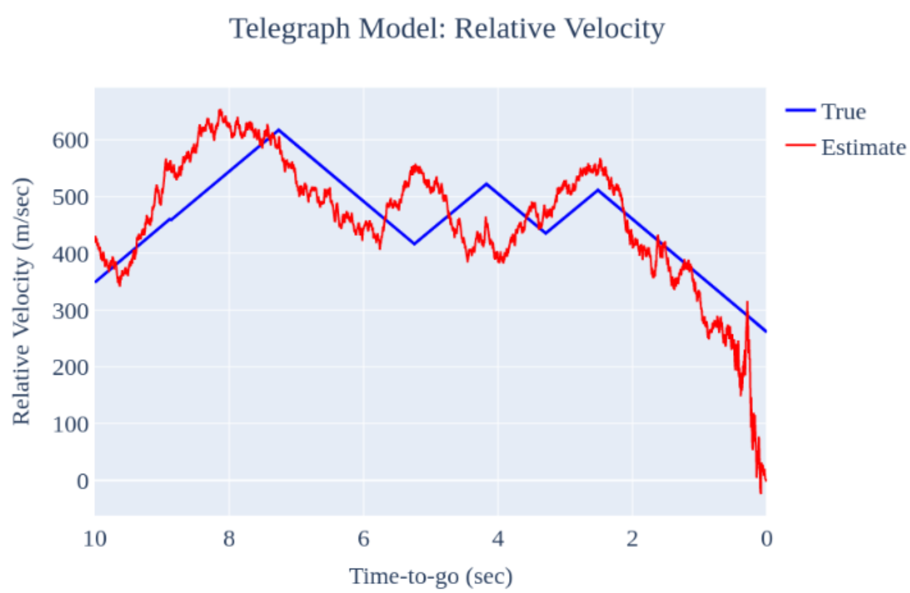

Figure 12: Telegraph velocity

6.3 Analysis of the Error Variance

To further analyze the performance of the filter under the telegraph model, the RMSE in state estimates is analyzed for a Monte Carlo simulation with 10,000 realizations. Figure 13 plots the actual versus a priori RMS position error, and Figure 14 plots the actual versus a priori RMS velocity error. The RMSE produced over the telegraph model's Monte Carlo simulation tracks the a priori RMSE in both position and velocity nearly as well as the RMSE produced over the Gauss-Markov model's Monte Carlo simulation. By taking the filter designed for the Gauss-Markov process and testing it in simulation driven by the telegraph model with different process dynamics, the filter's implementation is validated.

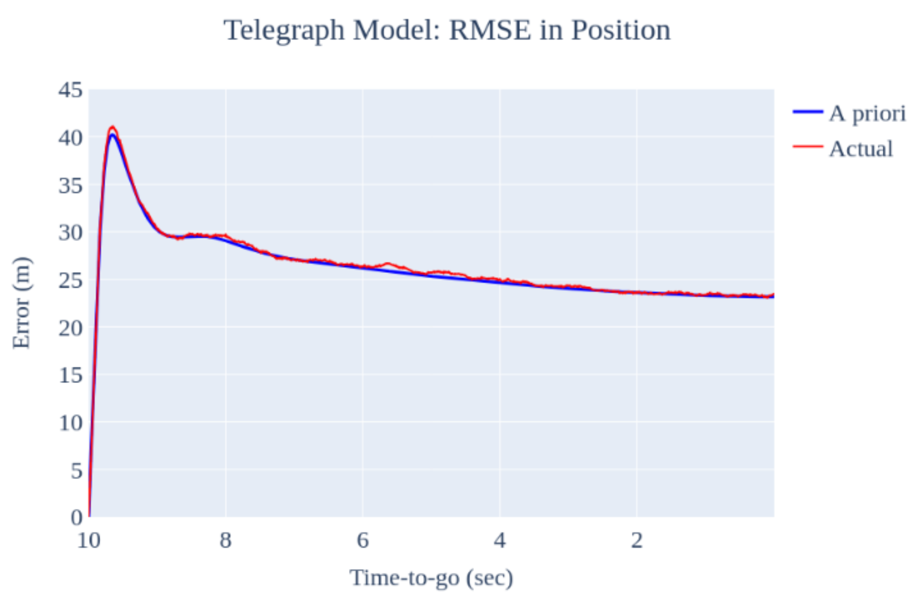

Figure 13: RMS error in position in the telegraph model.

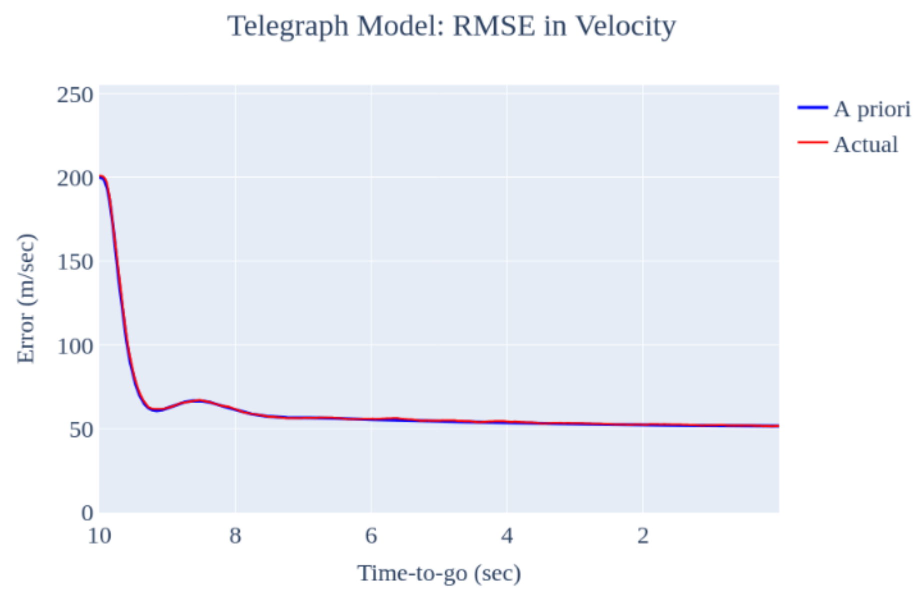

Figure 14: RMS error in velocity in the telegraph model.

7 Conclusion

A continuous-time Kalman Filter is implemented to estimate the system state in a missile intercept scenario. The filter is initially implemented by modeling the system as a Gauss-Markov process driven by a random forcing function with exponential correlation. The filter is shown to accurately track the relative position, relative velocity, and target acceleration states for one realization of the simulation using the Gauss-Markov model. Through a 10,000-realization Monte Carlo simulation, the RMSE in each state is shown to match the a priori values, thus verifying that the actual performance of the filter over the 10,000 realizations matches its theoretical performance. To test the filter's robustness to process and noise model errors, the Gauss-Markov model is replaced with a telegraph model. The target acceleration is modeled as a telegraph signal constructed to match the Gauss-Markov process in its mean and autocorrelation. The filter performance is then tested in a simulation driven by the telegraph model instead of the Gauss-Markov model. The filter is shown to accurately track the relative position and velocity between the pursuer and target for one realization of the telegraph model simulation. For a 10,000-realization Monte Carlo simulation, the RMSE in the filter's state estimates nearly matches the a priori RMSE, just as in the Gauss-Markov simulation. Therefore, the filter's performance is verified under two different process models, the Gauss-Markov model and the telegraph model, validating its implementation.