# 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.

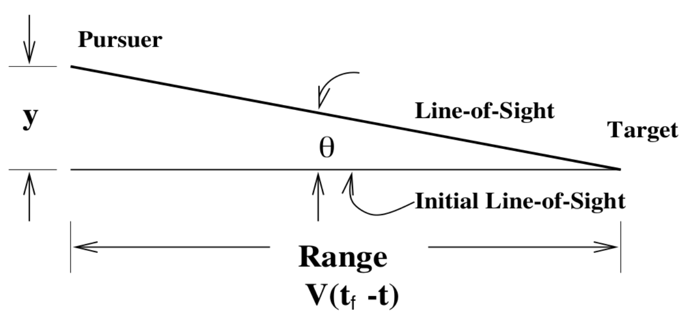


Figure 1: Missile Intercept Illustration.

## 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, $\tau$, 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, $\theta$. 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 to simulate the above model.

$$
\begin{array}{cccc}
V_c = 300\frac{ft}{sec} & E[a_T^2] = (100\frac{ft}{sec^2})^2 & t_f = 10sec & R_1 = 15 \times 10^{-6}\frac{rad^2}{sec} \\
R_2 = 1.67 \times 10^{-3}\frac{rad^2}{sec^3} & & \tau = 2sec & b = 1.52 \times 10^{-2}
\end{array}
$$

# 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

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

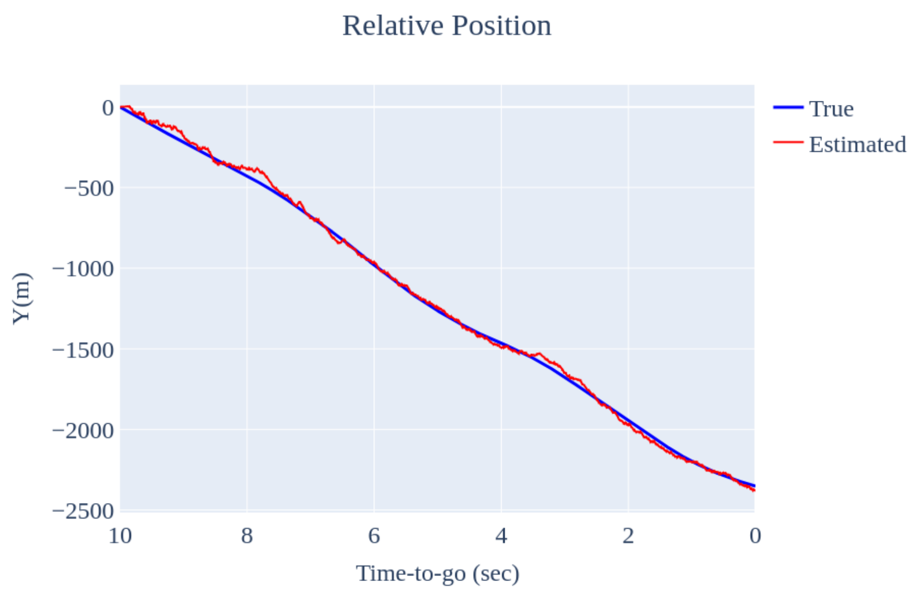


Figure 2: The true and estimated relative distance along y between the pursuer and target.

The true states are propagated via Euler integration using (6), and a noise sequence is generated to provide the simulation values of $w_{a_T}$. For each discrete-time instant in the simulation, the sensor measurement is simulated using (7).

### 5.2 Analysis of One Realization

### 5.3 Analysis of the Error Variance

## 6 Filter Robustness

## 7 Conclusion

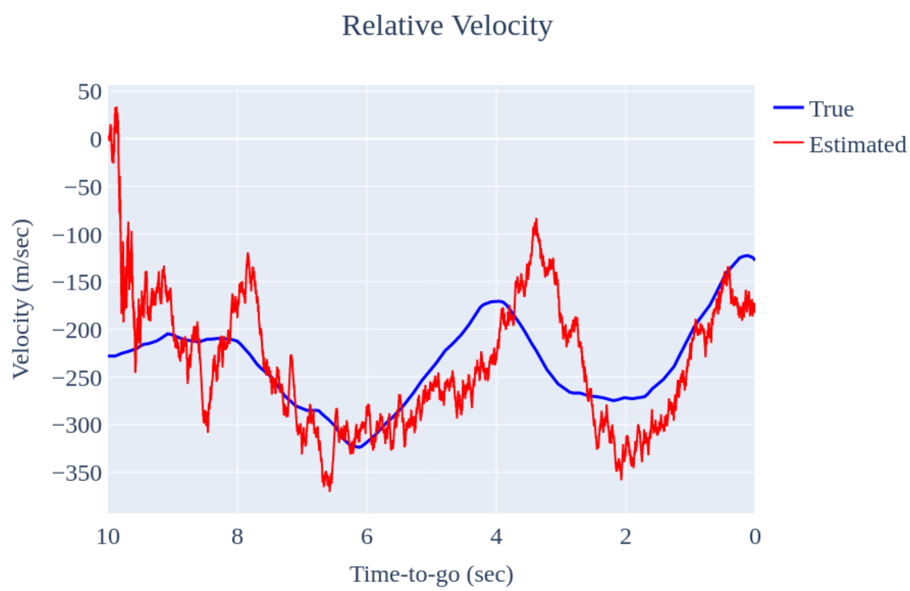


Figure 3: The true and estimated relative velocity along y between the pursuer and target.

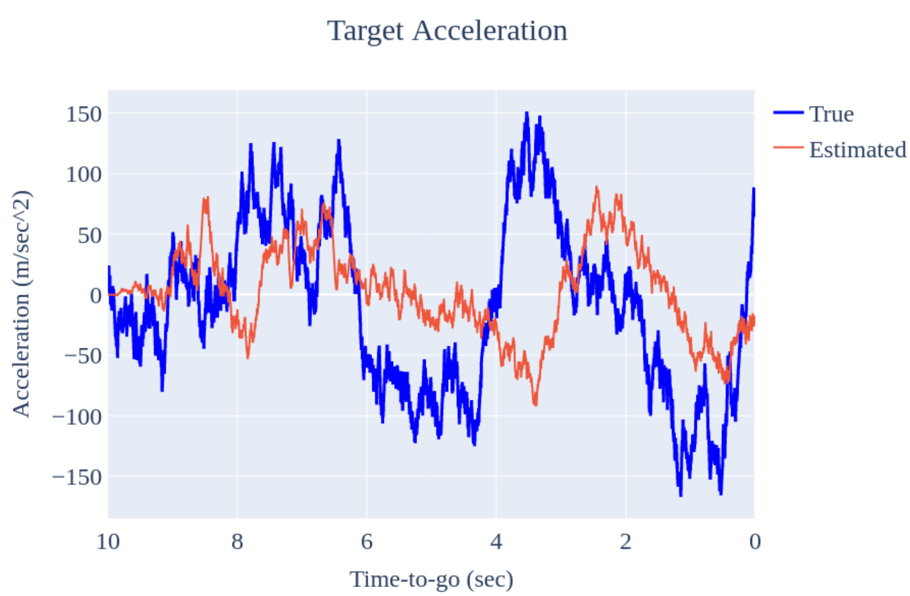


Figure 4: The true and estimated target acceleration.

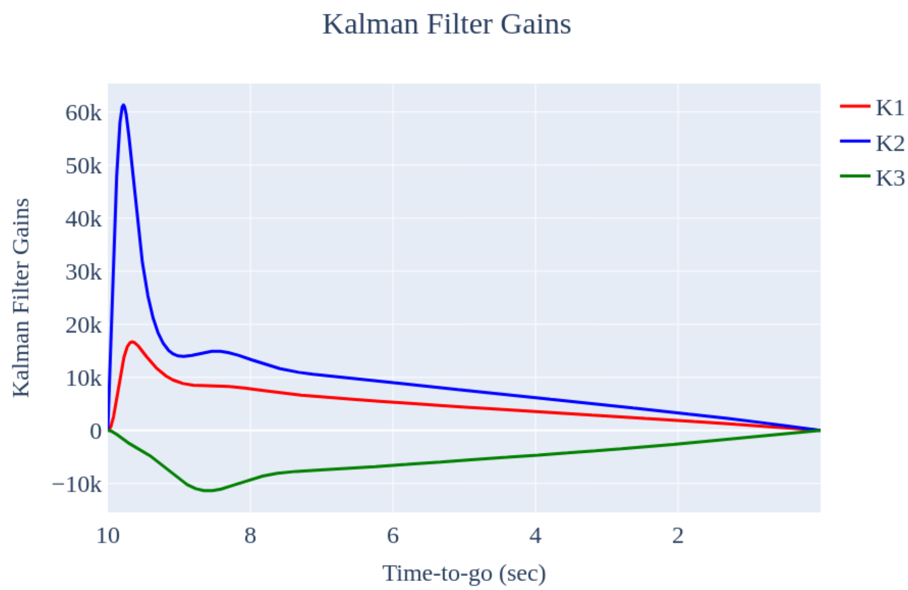


Figure 5: The Kalman gains.

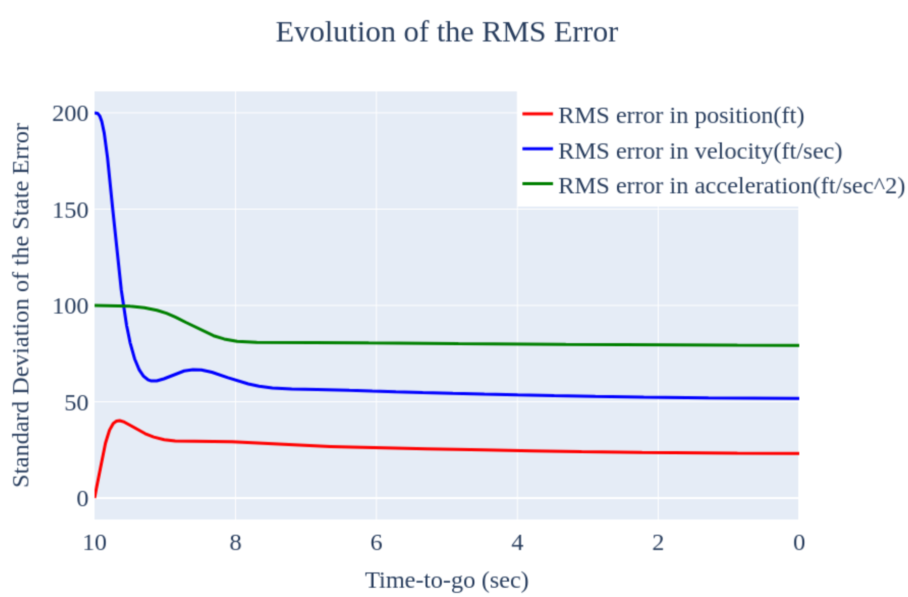


Figure 6: The evolution of the RMS error.

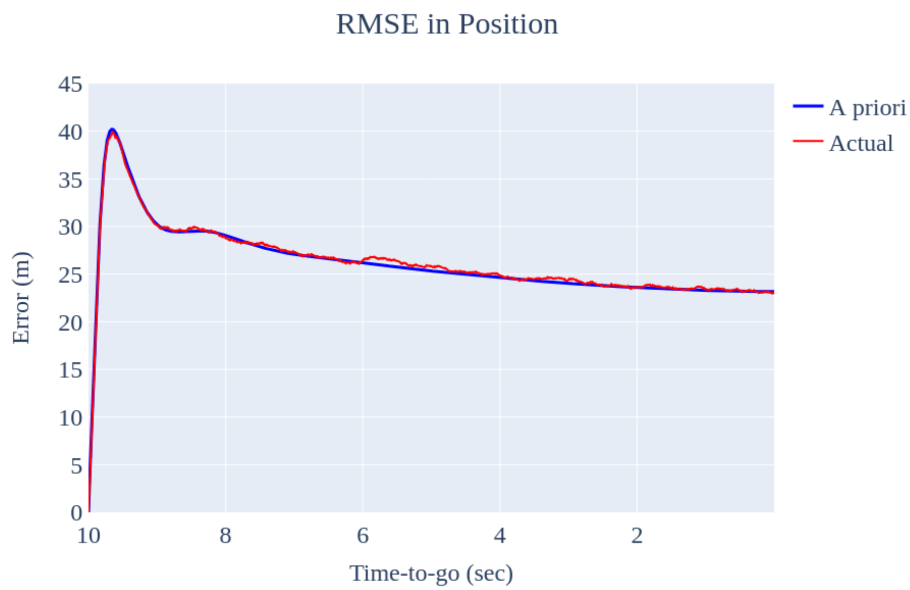


Figure 7: RMS error in position.

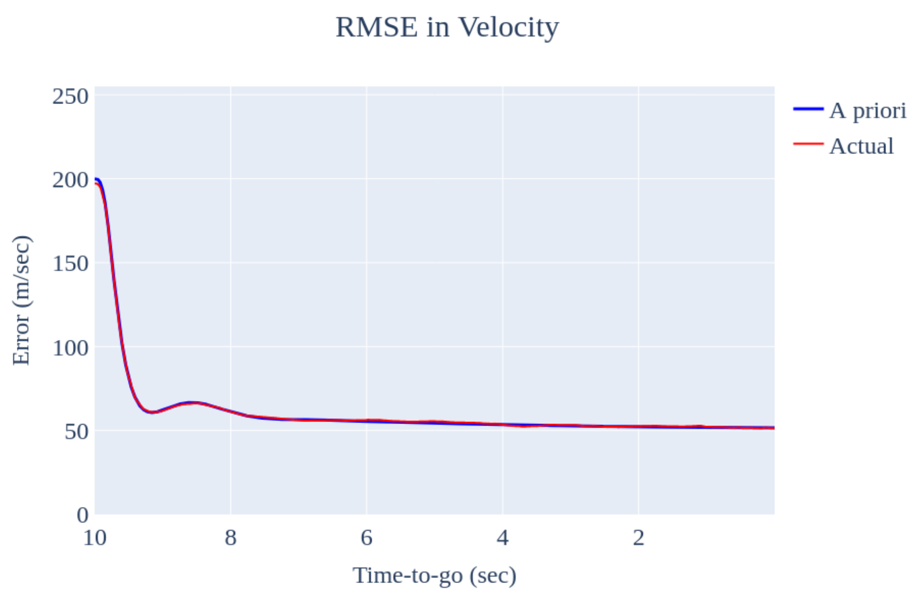


Figure 8: RMS error in velocity.

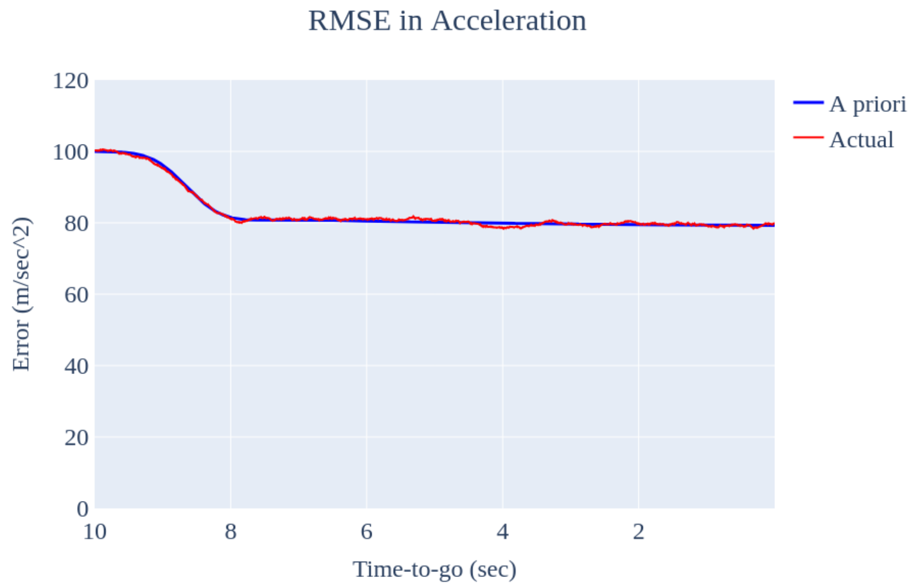


Figure 9: RMS error in acceleration.

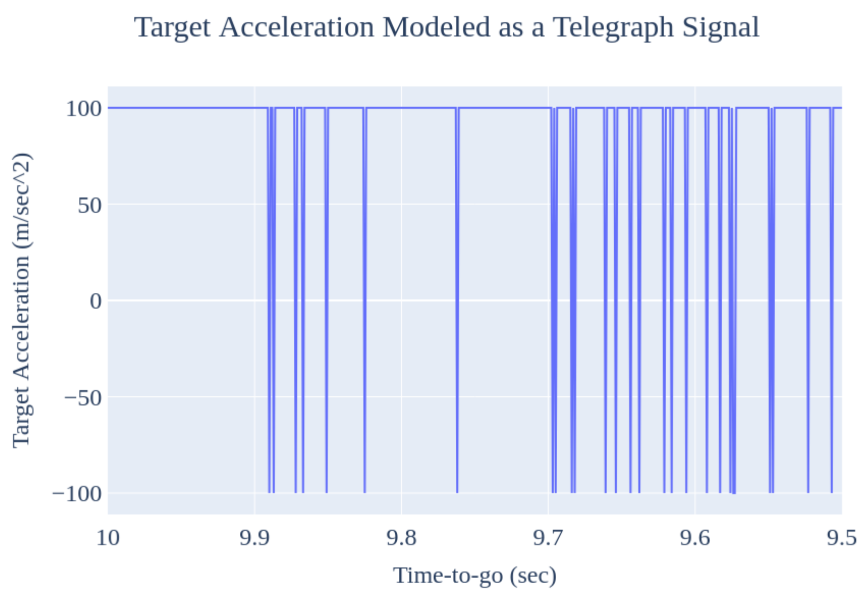


Figure 10: A close-up view of the target acceleration as modeled by the telegraph signal.

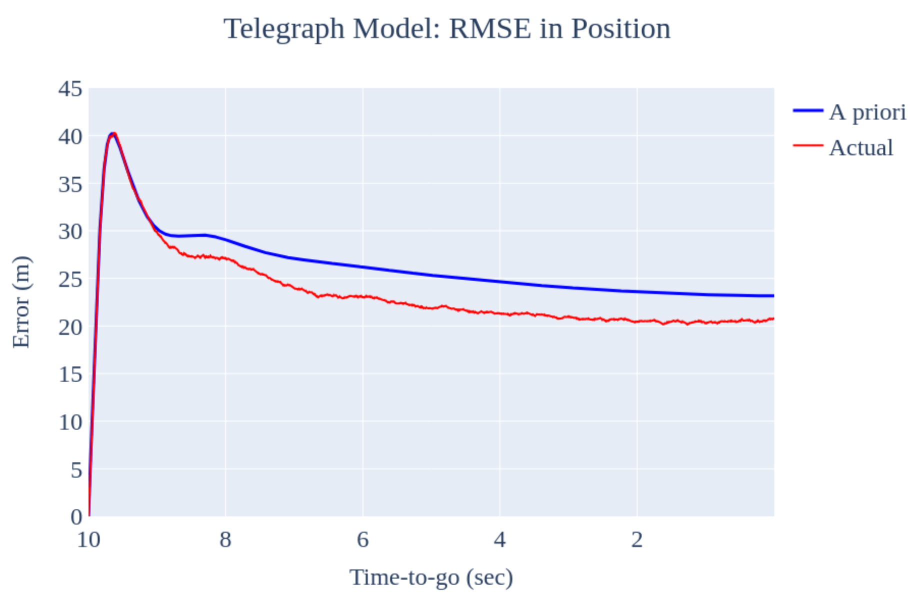


Figure 11: RMS error in position in the telegraph model.

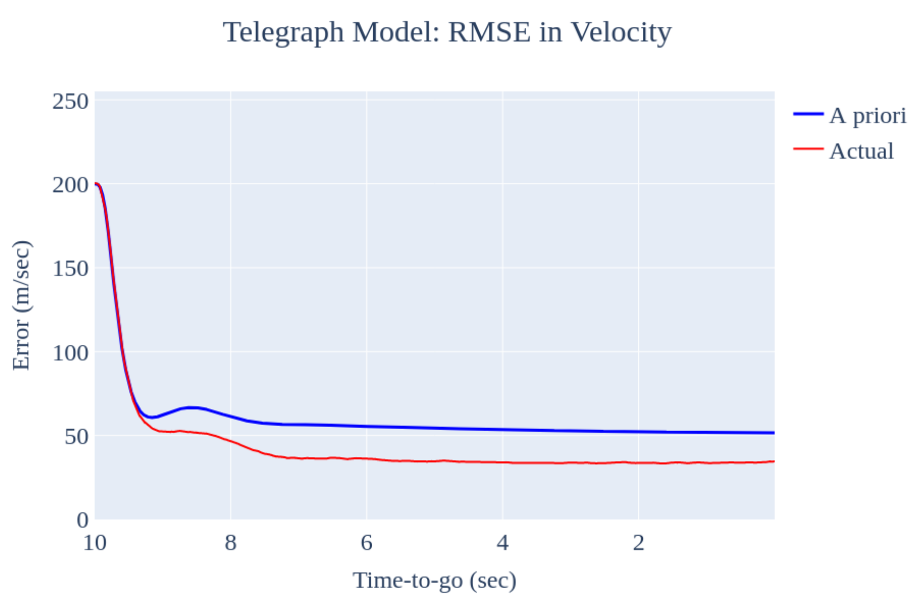


Figure 12: RMS error in velocity in the telegraph model.