AERSP 497 Autonomy

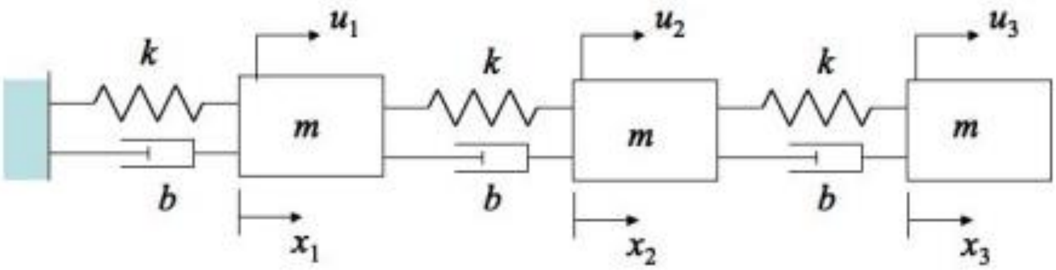
Project 1

Name	Contribution	Contribution %	Honor Statement Sign
Nicholas Giampetro	Code & Report	34	NG
Jackson Fezell	Code & Report	33	JF
Ankit Gupta	Code & Report	33	AG

Introduction

The project at hand delves into the exploration, simulation, and estimation of a three-body spring-mass damper system, which is defined and governed by certain equations of motion, matrices, and state vectors. The state-space representation of the system used is given by:

$$\dot{x} = A_c x + B_c u$$



where

$$A_c = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -2 & 1 & 0 & -1.6 & 0.8 & 0 \\ 1 & -2 & 1 & 0.8 & -1.6 & 0.8 \\ 0 & 1 & -1 & 0 & 0.8 & -0.8 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

The Kalman Filter is a process used to denoise a system that has non-perfect input data. Typically, the data is collected from sensors with some amount of unavoidable error, which can lead to an outcome that was slightly different from the predicted one. The filter is meant to provide an accurate guess as to how the system will evolve over time using previous measurement data. It can also predict derivatives of the motion of a system using error and measurements from the position of system. The result is a solution that closely resembles the predicted perfect guess, but better depicts the real-world scenario.

For our project, the "sensor noise" is simply a random number generator implemented into the code to replicate the noise a sensor would produce.

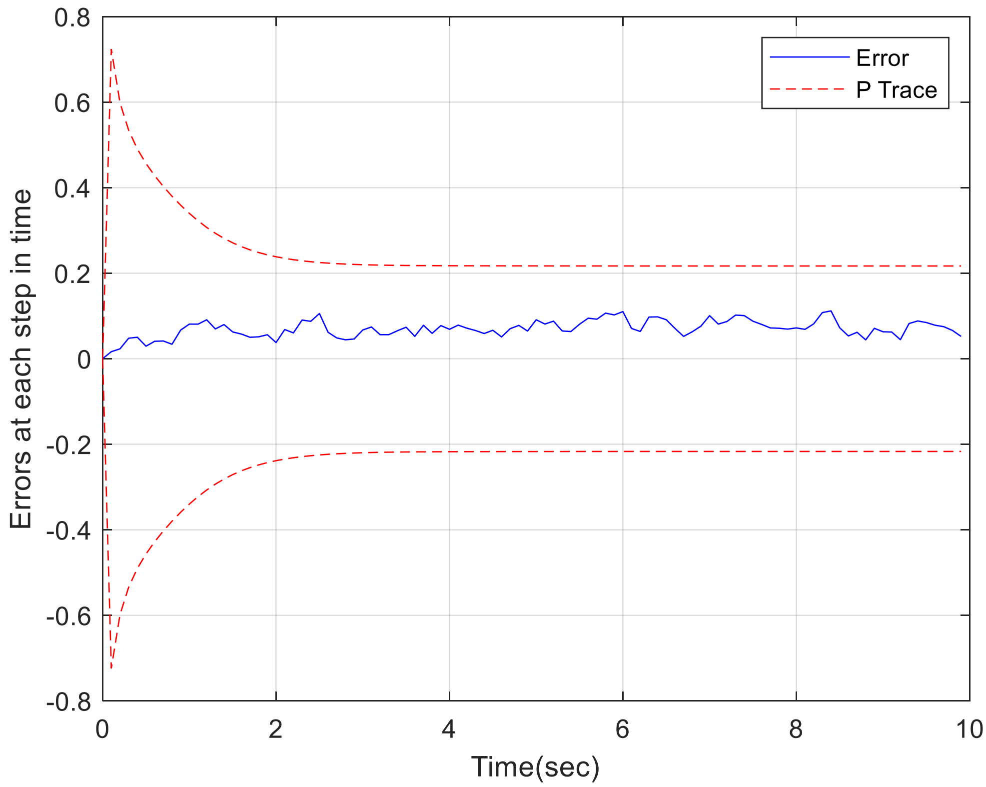
Part A

$$x_0 \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad u \sim \mathcal{N}(0, 0.2\boldsymbol{I})$$

Refer to the MATLAB code.

Part B

$$v \sim \mathcal{N}(0,0,1)$$

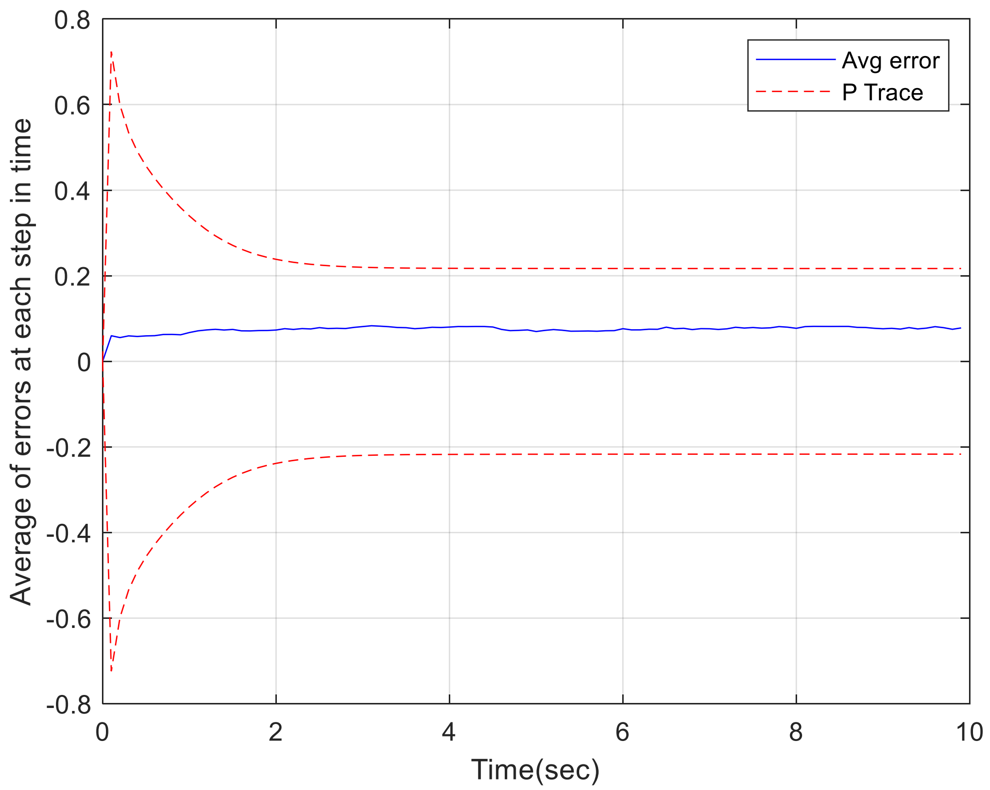


In Part B of the project, a 10-second simulation of the three-body spring-mass damper system was ran using the Kalman Filter algorithm, and the resulting plot shows the 2-norm of the estimated error and the square root of the trace of the covariance matrix.

The 2-norm of the estimated error shows how close the filter's estimates are to the actual values in the ideal solution. The greater the value of the 2-norm, the worse that the filter's prediction is. The square root of the trace of the covariance matrix shows the overall uncertainty that the filter has in its estimated solution values. The greater the value of this calculation, the greater the uncertainty. These values are used together to validate the accuracy of the Kalman Filter on the modeled dynamic system.

As seen in the plot, the 2-norm value starts at zero due to the first measurement being an initial condition but varies back and forth throughout the simulation. This suggests the precision in the filter's estimates continuously changes, but typically stays below 0.1 units. For the same reasons as the 2-norm value, the uncertainty of the estimates also starts at zero. The filter then becomes very uncertain in the beginning of the simulation, but quickly decreases and remains around 0.2 units. There is a large difference between the two measures, meaning that the implemented Kalman Filter is more accurate with its estimations, but less certain that this accuracy is correct. A smaller uncertainty would show that the Kalman Filter is able to model the dynamics of the given three-body spring-mass damper system both precisely and accurately.

Part C



This part of the project incorporates a Monte Carlo simulation to better represent the modeling of the three-body spring-mass damper system. A Monte Carlo simulation is used when probability or randomness is involved in a calculation to get the outcome of a model. The simulation consists of running the same algorithm many times over and taking an average of each result. This provides the most accurate representation of how the algorithm model will perform on average.

In the plot of the Monte Carlo simulation, the average 2-norm of the estimation error and the average square root of the trace of the covariance matrix are presented together. Observing the plot, it can be seen that the estimation error changed significantly, while the uncertainty is seemingly identical to that of Part B.

The error in the estimations were averaged out over 50 iterations to have an average value of about 0.1 units throughout the simulation. The uncertainty did not change from the single iteration ran in Part B. This is either due to the filter’s uncertainty being the same at each time step for each iteration, or the first iteration had very similar results to the average. Given the average values from the simulation, the Kalman Filter algorithm implemented on the dynamic system produces both accurate and precise results, especially after the two second mark.

Part D

Saving the square root of the trace of the covariance matrix at each time step during simulations helps to keep a record of how certain or uncertain the filter is about its estimations over the time period. This data allows us to visualize how well the filter is performing, especially in various scenarios or when things don’t go as expected. In these instances, it can be crucial for making improvements, spotting issues, or ensuring that it works reliably in different situations. So, while it might seem like extra work and computing power to keep this data, it provides valuable insights which can help us to refine and validate the filter's design and implementation.