MATH 596 MINI PROJECT 3

CAMERON HARRIS, GRANT HOLMES, MALENA SCHOENI & TRI PHAM

Introduction

This mini project serves as an implementation practice for a data assimilation technique (the Ensemble Kalman Filter) and its application to the chaotic Lorenz '63 model. We will provide a brief outline of our approach and observation below.

Implementation

4^{th} order Runge-Kutta method to solve the Lorenz '63 equations numerically

The Lorenz system is a system of ordinary differential equations that has become a standard example of sensitive dependence on initial conditions; that is, having chaotic solutions for certain parameter values and initial conditions. The system also exhibits a set of chaotic solutions known as the "Lorenz attractor," the collection of trajectories for different starting points tends to resemble a butterfly.

The Lorenz system includes three ordinary differential equations:

$$\frac{dx}{dt} = \sigma(y - x)$$
$$\frac{dy}{dt} = x(\rho - z) - y$$
$$\frac{dz}{dt} = xy - \beta z$$

The parameters σ, ρ, and β are assumed to be positive. Their usual values are:

$$\sigma = 10$$
$$\beta = \frac{8}{3}$$
$$\rho = 28$$

Next, we use the 4^{th} order Runge-Kutta method to sovle the Lorenz '63 equations with given time step of h=0.01. Without the loss of generality, we have the initial value problem specified as follows:

$$\frac{dy}{dt} = f(t, y), with\ y(t_0) = y_0$$

Date: November 18, 2020.

Our task is to find value of unknown function y of time t, so we will use the following slope approximations to estimate the slope at different t $\in [t_0, t_1, ..., t_n]$. In particular,

k_0 is the increment based on the slope at the beginning of the interval using y.

k_1 is the increment based on the slope at the midpoint of the interval using $y + h\frac{k_0}{2}$.

k_2 is the increment based on the slope at the midpoint of the interval using $y + h\frac{k_1}{2}$.

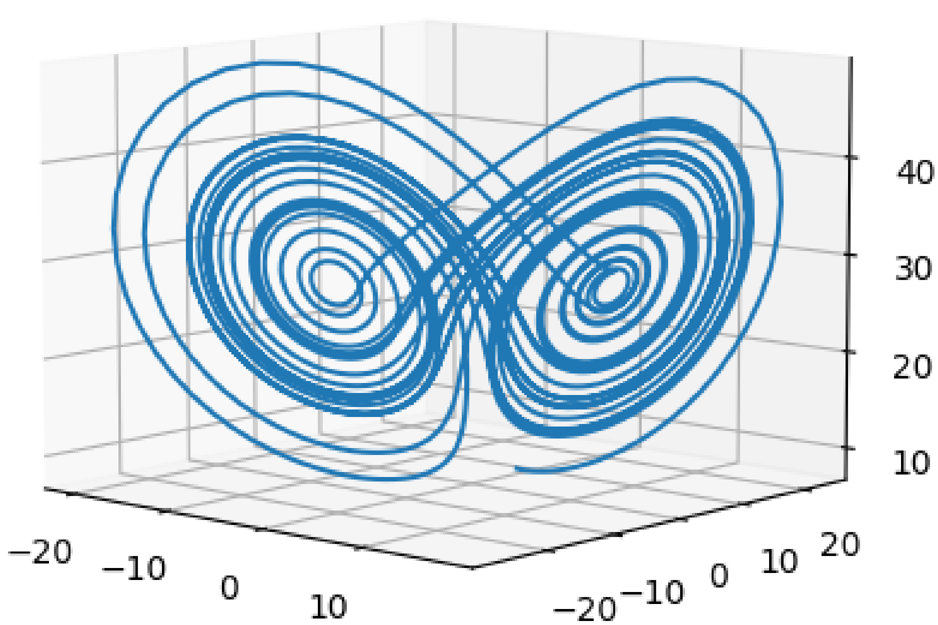
k_3 is the increment based on the slope at the end of the interval using $y + hk_2$.

$$k_0 = hf(t_i, y_i)$$
$$k_1 = hf(t_i + \frac{h}{2}, y_i + \frac{k_0}{2})$$
$$k_2 = hf(t_i + \frac{h}{2}, y_i + \frac{k_1}{2})$$
$$k_3 = hf(t_i + h, y_i + k_2)$$

Thus, we can then form the 4^{th} order Runge-Kutta approximation y_{i+1} for t_{i+1} where:

$$y_{i+1} = y_i + \frac{1}{6}h(k_0 + 2k_1 + 2k_2 + k_3)$$
$$t_{i+1} = t_i + h$$

With $[x, y, z] = [2, 4, 8]$ and default values of σ, ρ, and β, we have the solutions to the Lorenz attractor using 4^{th} order Runge Kutta below



Ensemble Kalman Filter for linear observation operators

We have a combination of a physical model and a data model where both have some uncertainty. Let physical model be denoted by

$v_{j+1} = \Psi(v_j) + \xi_j$ where $j = 0, 1, ..., \xi_j \sim \mathcal{N}(0, \Sigma), v_0 \sim \mathcal{N}(m_0, C_0)$

and data model be denoted by

$y_{j+1} = h(v_{j+1}) + \eta_{j+1}$ where $j = 0, 1, ..., h : \mathbb{R}^n \to \mathbb{R}^m, \eta_j \sim \mathcal{N}(0, \Gamma)$

We then apply EnKF to our system by applying prediction and update step for each ensemble up to N.

$$\text{Prediction} \begin{cases} \hat{v}_{j+1}^{(n)} = \Psi(v_j^{(n)}) + \xi_j^{(n)}, \ n = 1, ..., N \\ \hat{m}_{j+1} = \frac{1}{N}\Sigma_{n=1}^{N}\hat{v}_{j+1}^{(n)} \\ \hat{C}_{j+1} = \frac{1}{N-1}\Sigma_{n=1}^{N}(\hat{v}_{j+1}^{(n)} - \hat{m}_{j+1})(\hat{v}_{j+1}^{(n)} - \hat{m}_{j+1})^T \end{cases}$$

$$\text{Update} \begin{cases} S_{j+1} = H\hat{C}_{j+1}H^T + \Gamma \\ K_{j+1} = \hat{C}_{j+1}H^T S_{j+1}^{-1} \\ y_{j+1}^{(n)} = y_{j+1} + \eta_{j+1}^{(n)}, \ n = 1, ..., N \\ v_{j+1}^{(n)} = (I - K_{j+1}H)\hat{v}_{j+1}^{(n)} + K_{j+1}y_{j+1}^{(n)}, \ n = 1, ..., N \end{cases}$$

Experiments and observations

After applying the Ensemble Kalman Filter to several different hypothetical scenarios we found that, on average, the filter performs best with a small time step, all three variables being observed, and at least 15 ensemble members. This is consistent with our intuition that having more data will generally improve results. If we only observe one or two variables, there will be important information not being considered, thus decreasing the overall accuracy. Listed below are graphs of our hypothetical scenarios.

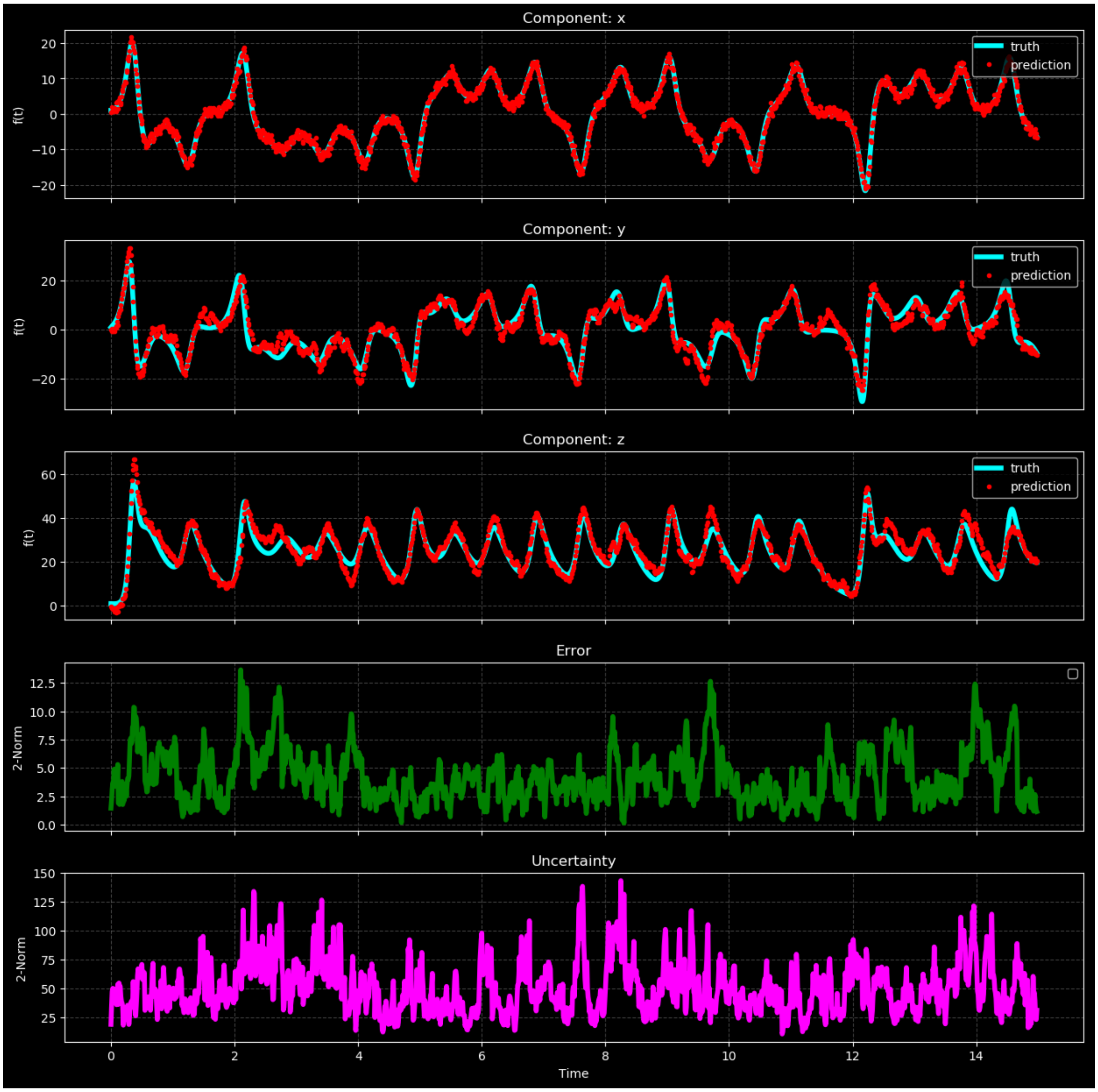


Fig1. Observation for one single variable x.

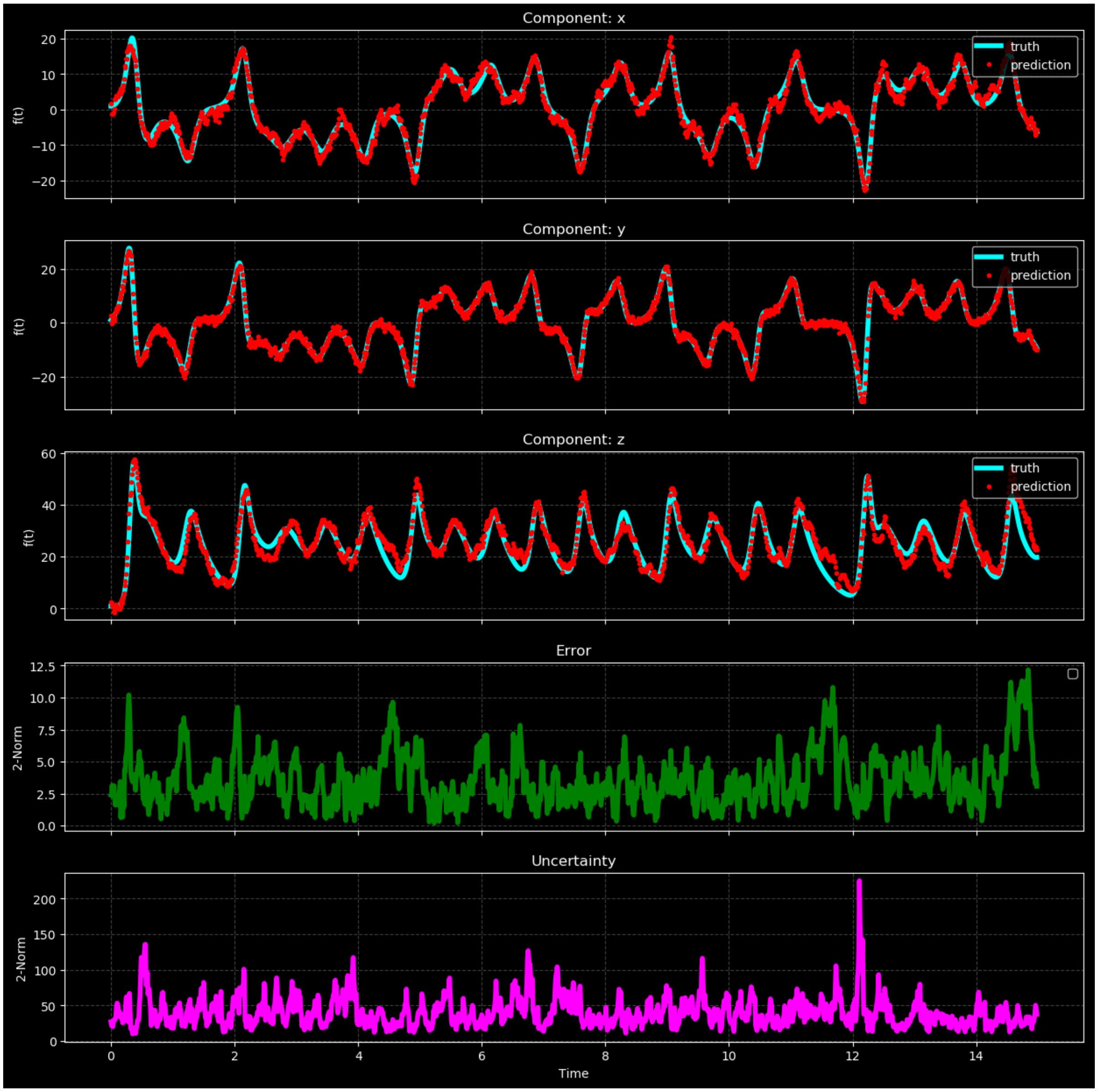


Fig2. Observation for one single variable y.

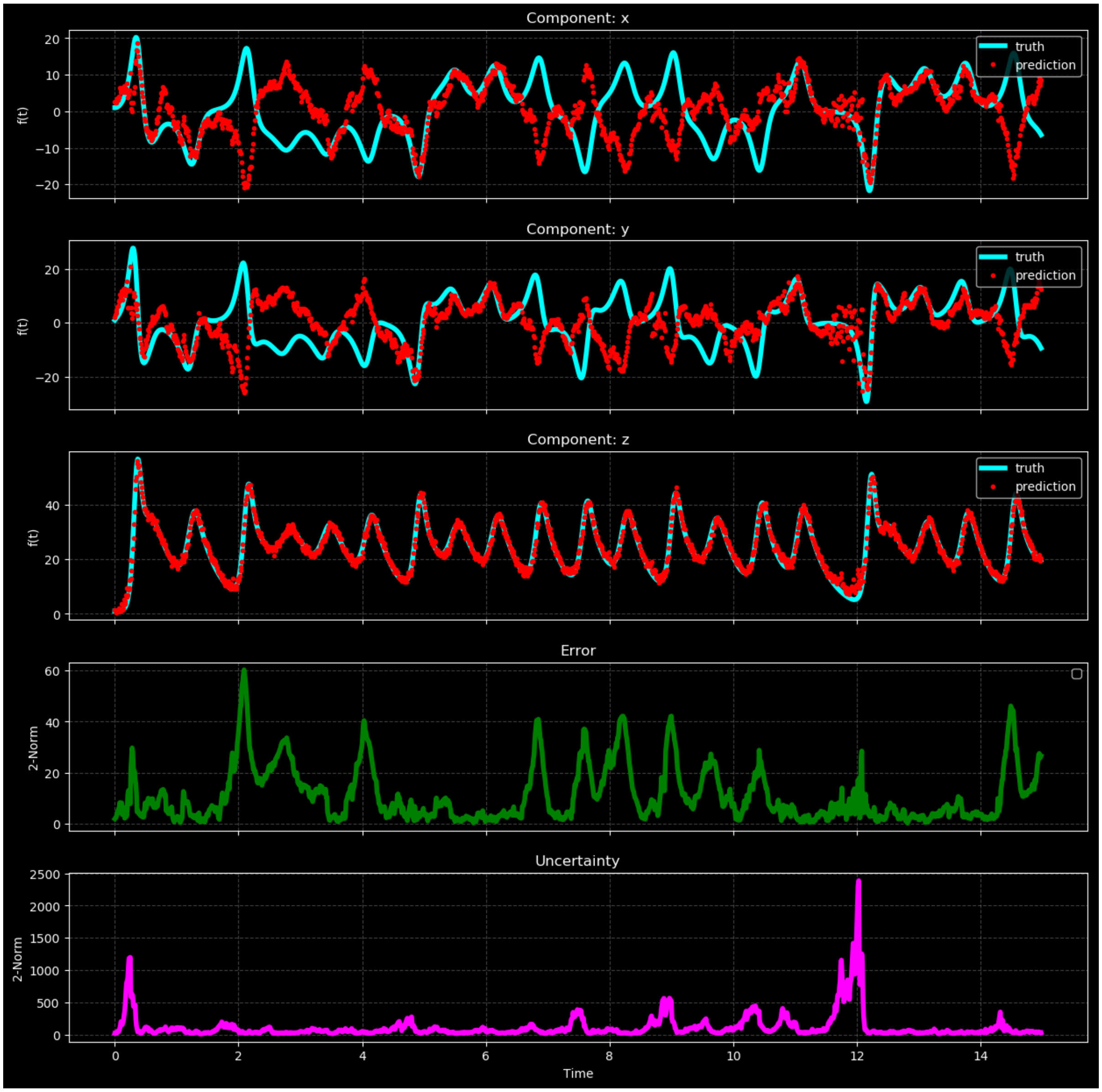


Fig3. Observation for one single variable z.

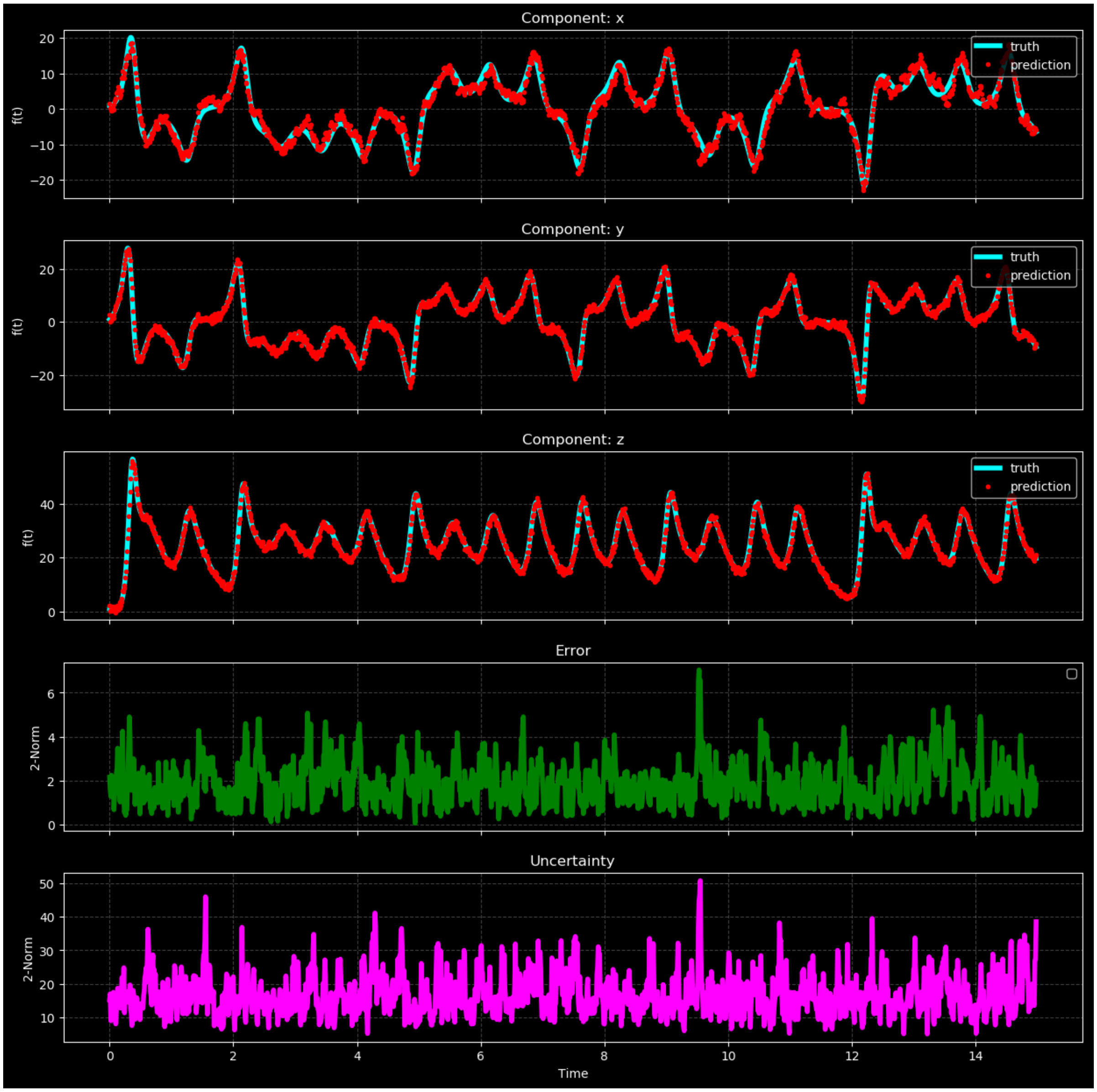


Fig4. Observation for two variables y and z.

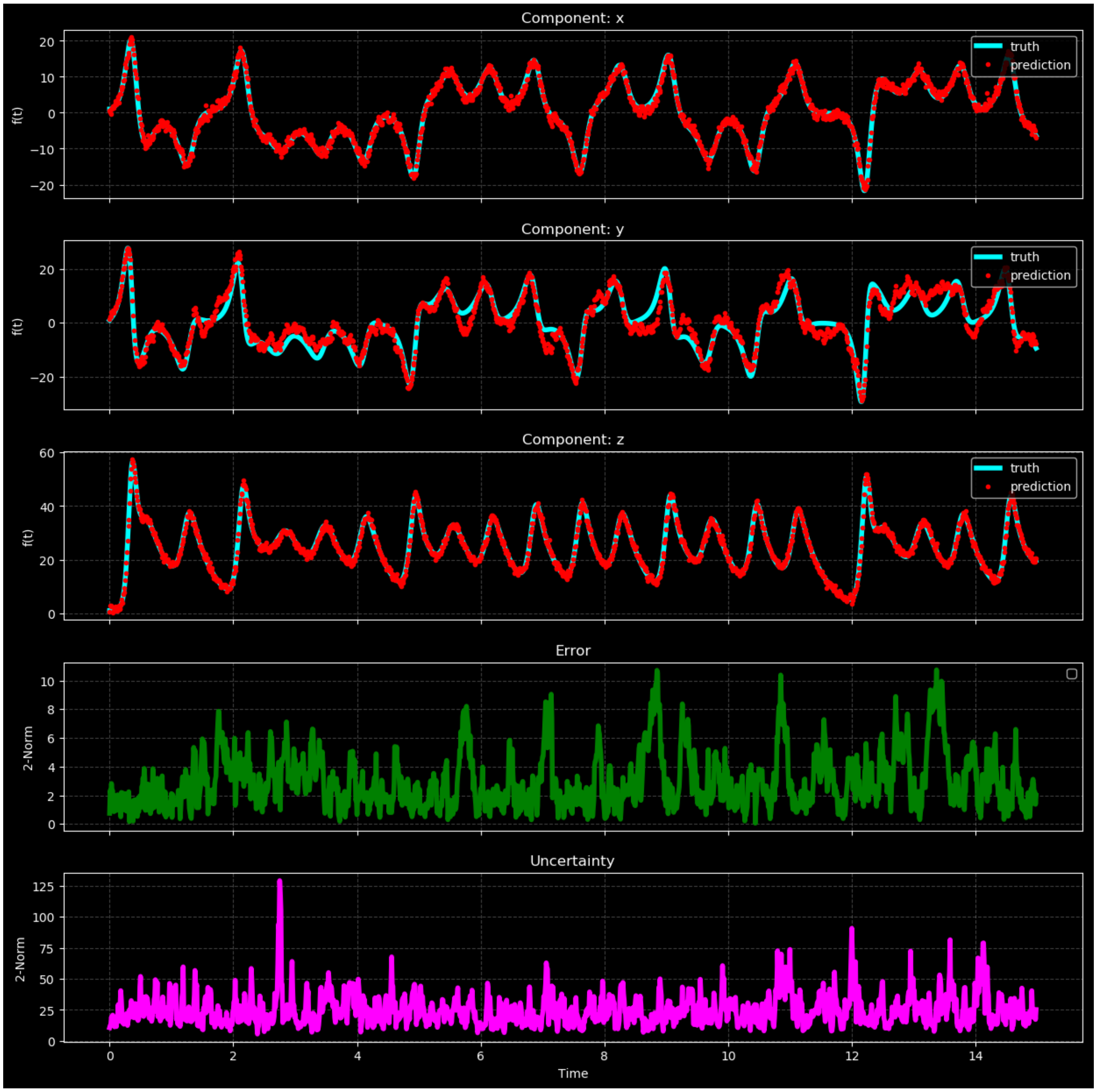


Fig5. Observation for two variables x and z.

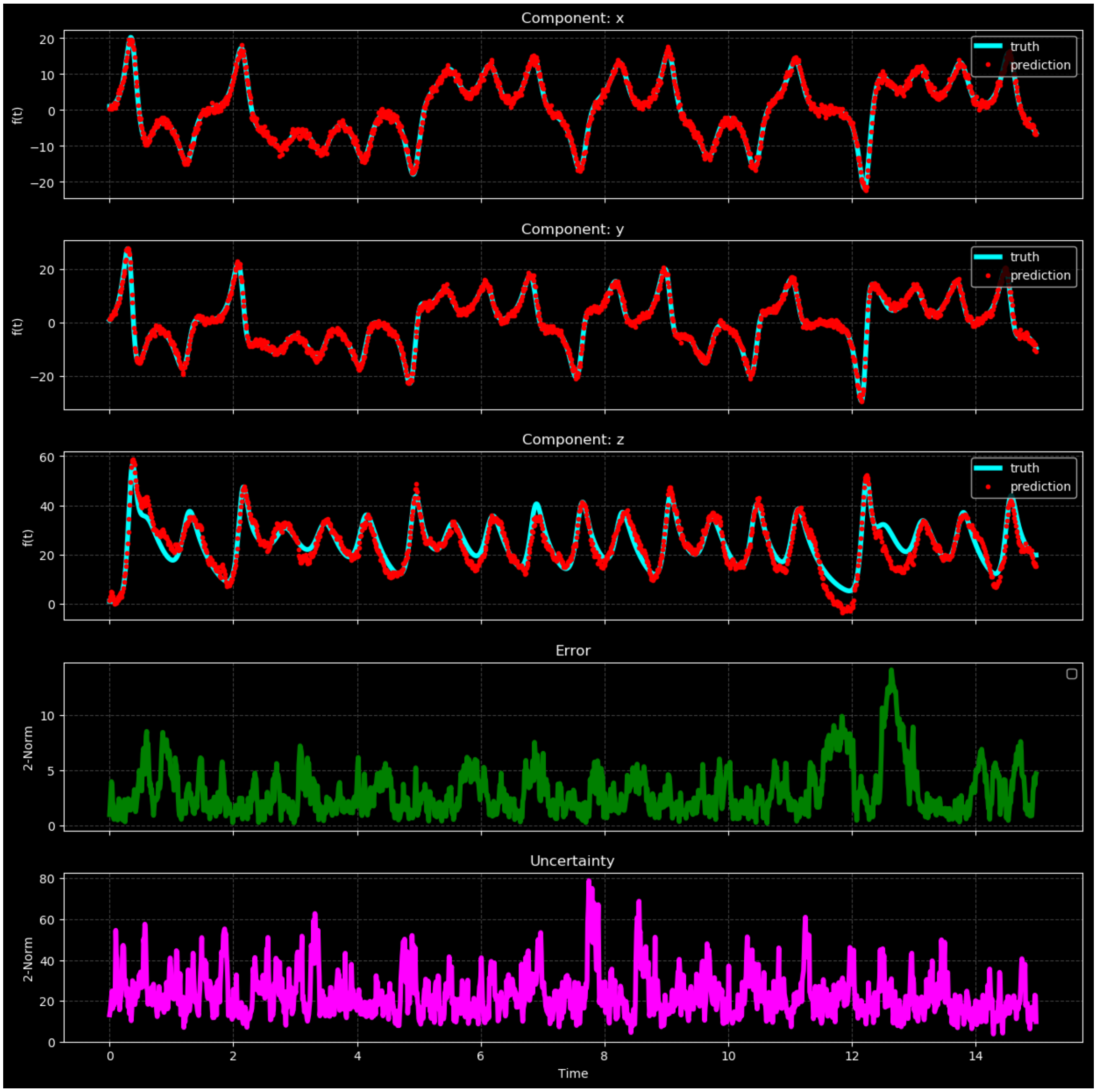


Fig6. Observation for two variables x and y.

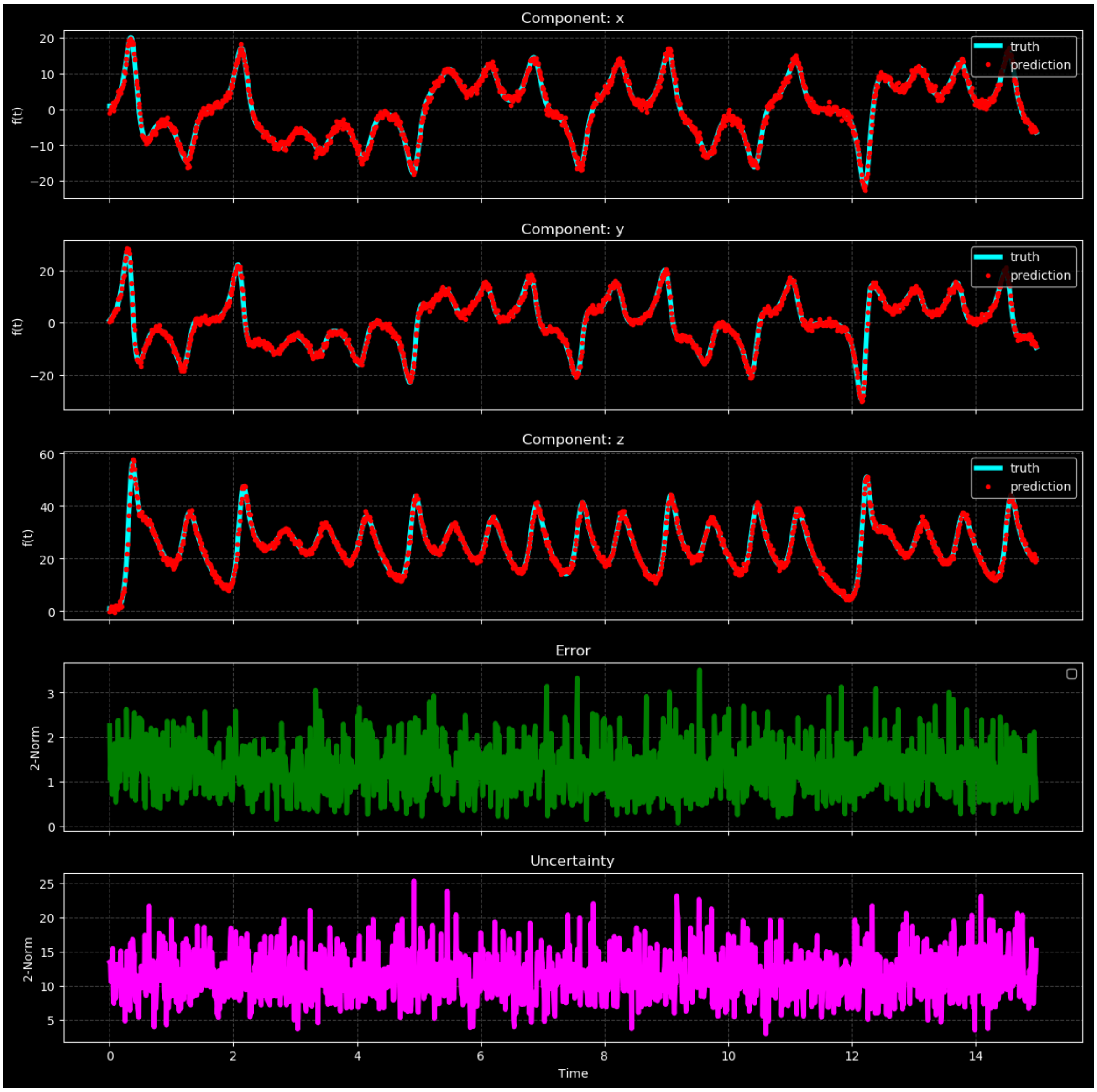


Fig7. Observation for all three variables x, y, and z.

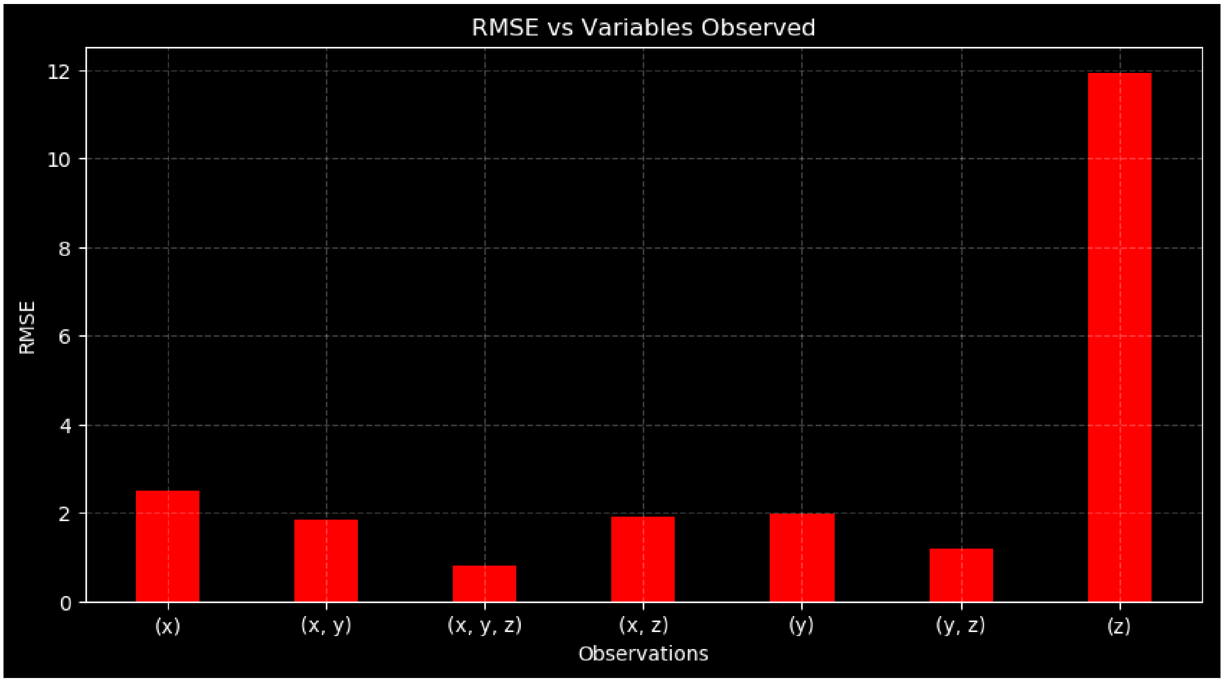


Fig8. RMSE metric results for each of the 7 above scenarios.

Our implementation can be found at github.com/gholmes829/Math-596-Project-3.

References

[1] en.wikipedia.org/wiki/Lorenz_system

[2] people.sc.fsu.edu/∼jburkardt/py_src/lorenz_ode/lorenz_ode.html

[3] en.wikipedia.org/wiki/Runge–Kutta_methods

[4] geeksforgeeks.org/runge-kutta-4th-order-method-solve-differential-equation