EKF-ST: Simultaneous State Estimation and Sparse Equation Learning

MAE 6760 Project Report

Zhiheng Chen

Abstract: In this project, I devised a new algorithm that enables simultaneous state estimation and sparsity-promoting equation learning under process and sensor noises. My algorithm combines the extended Kalman filter and sequential thresholding techniques, and thus it is named as EKF-ST. I implemented this algorithm on the Lorenz system to test its effectiveness.

1 Project Goal

The goal of the project is to construct the EKF-ST algorithm for general dynamical systems, and test it on the Lorenz system. The project considers an n-dimensional nonlinear dynamical system with additive Gaussian process noise:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{G}\boldsymbol{w}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state space vector, $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is the dynamic model, and $\boldsymbol{G}\boldsymbol{w}$ is the added process noise with the covariance of $\boldsymbol{w}$ denoted as $\boldsymbol{Q}$. The sensors give partial state measurements:

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v}$$

where $\boldsymbol{z} \in \mathbb{R}^m$ is the measurement, $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^m$ is the sensor model, and $\boldsymbol{v} \in \mathbb{R}^m$ is the sensor noise whose covariance is denoted as $\boldsymbol{R}$.

In this project, I assume that both the coefficients and the basis functions in the i-th equation of the system is not fully known, and thus we do not have a good model of the system at the start. The goal of EKF-ST is then to recursively learn the correct equation via sequential thresholding, aided by the estimation results from extended Kalman filter.

The EKF-ST algorithm will then be tested on the Lorenz system:

$$\begin{aligned}
\dot{x} &= -\sigma x + \sigma y + w_1 \\
\dot{y} &= \rho x - y - xz + w_2 \\
\dot{z} &= -\beta z + xy + w_3
\end{aligned}$$

with its first equation not fully konwn. The measurements are

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v}$$

Both full and partial state measurement cases will be explored.

2 The EKF-ST Algorithm

Given that the i-th equation is not fully known, we model it as a linear combination of candidate basis functions with coefficients θ_j:

$$f_i \approx \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_3 + \theta_5 x_1 x_2 + \theta_6 x_2 x_3 + \dots$$

Then the augmented state is

$$\boldsymbol{x}_a = \begin{bmatrix} x_1 & \dots & x_n & \theta_1 & \dots & \theta_b \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{x}^T & \boldsymbol{\theta}^T \end{bmatrix}^T$$

where b stands for the total number of candidate basis functions. Then the system dynamics becomes

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_i \\ \vdots \\ \dot{x}_n \\ \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_b \end{bmatrix} = \underbrace{\begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_3 + \theta_5 x_1 x_2 + \theta_6 x_2 x_3 + \dots \\ \vdots \\ f_n(\boldsymbol{x}) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\boldsymbol{f}_a(\boldsymbol{x}_a)} + \boldsymbol{G}_a \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{w}_\theta \end{bmatrix}$$

Euler discretization gives

$$\boldsymbol{x}_{a,k+1} = \boldsymbol{x}_{a,k} + \boldsymbol{f}_a(\boldsymbol{x}_{a,k})\Delta t + \boldsymbol{G}_a \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{w}_\theta \end{bmatrix} \Delta t$$

Then the discrete Jacobian matrix $\boldsymbol{F}_k \in \mathbb{R}^{(n+b)\times(n+b)}$ is

$$\boldsymbol{F}_k = \frac{\partial}{\partial \boldsymbol{x}_{a,k}} \left(\boldsymbol{x}_{a,k} + \boldsymbol{f}_a(\boldsymbol{x}_{a,k})\Delta t \right)$$

Theoretically, we can then apply ordinary EKF to estimate the augmented state. However, in the presence of multiple candidate basis functions with many of them being spurious, directly using EKF may cause false positives (i.e., some of the basis functions that should have zeros coefficients can be mistakenly excited).

This can be mitgated by implementing a sequential thresholding process. Specifically, we set a threshold knob λ_{ST} that is a positive small number. Denote the posterior state estimate and covariance at step k as $\hat{\boldsymbol{x}}_{a,k+1|k+1}$ and $\boldsymbol{P}_{a,k+1|k+1}$. If we find any element $\hat{x}_{a,i}$ in $\hat{\boldsymbol{x}}_a$ such that $|\hat{x}_{a,i}| < \lambda_{ST}$, then we set $\hat{x}_{a,i}$ and the covariances in the i-th row and column of $\boldsymbol{P}_{a,k+1|k+1}$ to zero, and we set the i-th diagonal element of $\boldsymbol{P}_{a,k+1|k+1}$ (i.e., the variance of $\hat{x}_{a,i}$) to a small number. After thresholding, we proceed with the next EKF loop. Therefore, the EKF-ST algorithm can be written as

1. Pick candidate basis functions and construct $\boldsymbol{x}_a(\boldsymbol{x}, \boldsymbol{\theta})$ and $\boldsymbol{f}_a(\boldsymbol{x}_a)$
2. Pick λ_{ST}, $\hat{\boldsymbol{x}}_{a,0|0}$ and $\boldsymbol{P}_{a,0|0}$

3. for $k = 0, 1, \ldots$
   1) Compute $\boldsymbol{F}_k$
   2) Do EKF state and covariance predictions for $\hat{\boldsymbol{x}}_{a,k+1|k}$ and $\boldsymbol{P}_{a,k+1|k}$
   3) Compute the Kalman gain and find the posterior estimates $\hat{\boldsymbol{x}}_{a,k+1|k+1}$ and $\boldsymbol{P}_{a,k+1|k+1}$
   4) Find smallInds such that $|\hat{\boldsymbol{x}}_{a,k+1|k+1}(\text{smallInds})| < \lambda_{ST}$ and smallInds $> n$
   5) $\hat{\boldsymbol{x}}_{a,k+1|k+1}(\text{smallInds}) \leftarrow 0$; $\boldsymbol{P}_{a,k+1|k+1}(\text{smallInds}, :) \leftarrow 0$; $\boldsymbol{P}_{a,k+1|k+1}(:, \text{smallInds}) \leftarrow 0$
   6) $\boldsymbol{P}_{a,k+1|k+1}(\text{smallInds}, \text{smallInds}) \leftarrow 0.5$

   end

I keep a small value for the variances of the thresholded coefficients, leaving some room for them to recover if they are set to zero by mistake.

3 Numerical Experiments on the Lorenz System

I tested the EKF-ST algorithm on the Lorenz system (first equation unknown) with different candidate basis functions and state measurements. I also compared the results with the ordinary EKF and the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm [1]. My code is available at the EKF-ST GitHub Repository I created.

To generate measurements and obtain the true trajecory and parameters for error analysis, I ran an ODE45 simulation on the true Lorenz system with $\sigma = 10$, $\rho = 8/3$, $\beta = 28$, and initial condition $\boldsymbol{x}_0 = [1\ 1\ 1]^T$. The time steps for the simulation and EKF-ST are both 0.001 s. The process noise covariance is $\boldsymbol{Q} = \text{diag}\{0.1^2, 0.1^2, 0.1^2\}$, and the measurement noise variance for each state variable (if measured) is 1.

3.1 Full State Measurements with 1 and Linear Bases

The dynamics are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = \begin{bmatrix} \theta_1 + \theta_2 x + \theta_3 y + \theta_4 z \\ \rho x - y - xz + w_2 \\ -\beta z + xy + w_3 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{w_\theta} \end{bmatrix}$$

And the measurements are

$$\boldsymbol{z} = \underbrace{\begin{bmatrix} \boldsymbol{I}_{3\times 3} & 0_{3\times 4} \end{bmatrix}}_{\boldsymbol{H}_a} \boldsymbol{x}_a + \boldsymbol{v}$$

I set $\lambda_{ST} = 0.1$, $\hat{\boldsymbol{x}}_{a,0|0} = [1\ 1\ 1\ 10\ 10\ 10\ 10]^T$, and $\boldsymbol{P}_{a,0|0} = \text{diag}\{10, 10, 10, 30, 30, 30, 30\}$. Then I implemented the EKF-ST algorithm.

The coefficients converge to $\boldsymbol{\theta} = [0\ \ -10.2086\ \ 10.1173\ \ 0]^T$, and thus the learned equation is $\dot{x} = -10.2086x + 10.1173y$. The estimated trajectory and the true trajectory is shown in Fig. 1. The estimation errors for the state variables and the basis coefficients are shown in Fig. 2.

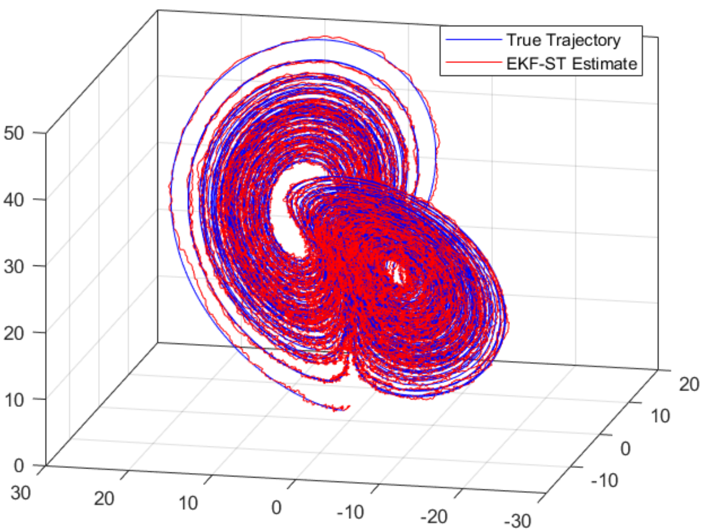

Figure 1: Experiment 1 true and estimated trajectories.

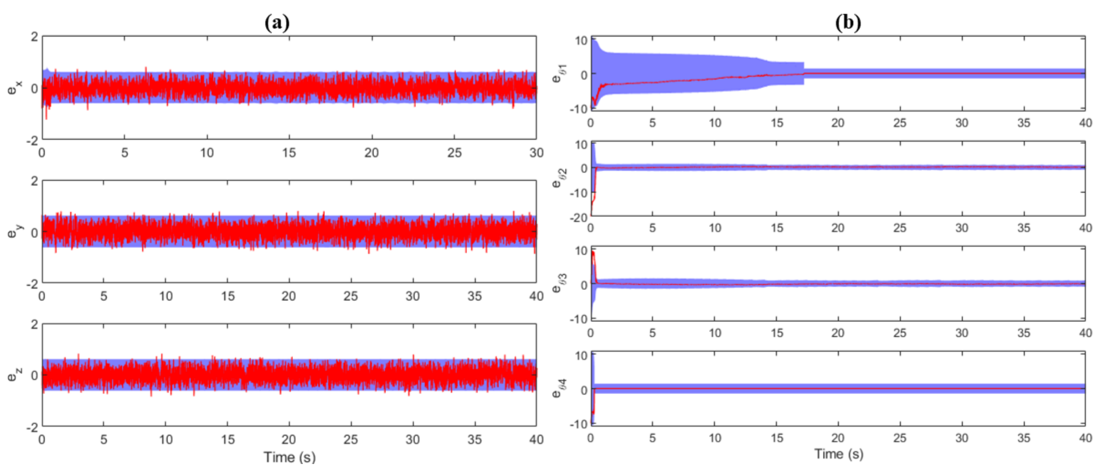

Figure 2: Experiment 1 error and 2-sigma bounds for (a) state estimations and (b) basis coefficient estimations.

From the figures, the estimation errors mostly lie within the 2-sigma bounds, implying that the filter is consistent.

3.2 Partial State Measurements with 1, Linear, and Cross-Term Bases

The dynamics are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_7 \end{bmatrix} = \begin{bmatrix} \theta_1 + \theta_2 x + \theta_3 y + \theta_4 z + \theta_5 xy + \theta_6 yz + \theta_7 xz \\ \rho x - y - xz + w_2 \\ -\beta z + xy + w_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{w_\theta} \end{bmatrix}$$

Only x and y are measured, and thus the measurements are

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{I}_{2\times 2} & 0_{2\times 8} \end{bmatrix} \boldsymbol{x}_a + \boldsymbol{v}$$

I set $\lambda_{ST} = 0.1$, $\hat{\boldsymbol{x}}_{a,0|0} = [1\ 1\ 1\ 10\ 10\ 10\ 10]^T$, and $\boldsymbol{P}_{a,0|0} = 10^3 \times \text{diag}\{1, 0.01, 0.01, 1, 1, 1, 1, 1, 1, 1\}$. Then I implemented the EKF-ST algorithm.

The coefficients converge to $\boldsymbol{\theta} = [0\ \ -10.8859\ 10.5487\ 0\ 0\ 0\ 0]^T$, and thus the learned equation is $\dot{x} = -10.8859x + 10.5487y$. The estimated trajectory and the true trajectory is shown in Fig. 3. The estimation errors for the state variables and the basis coefficients are shown in Fig. 4.

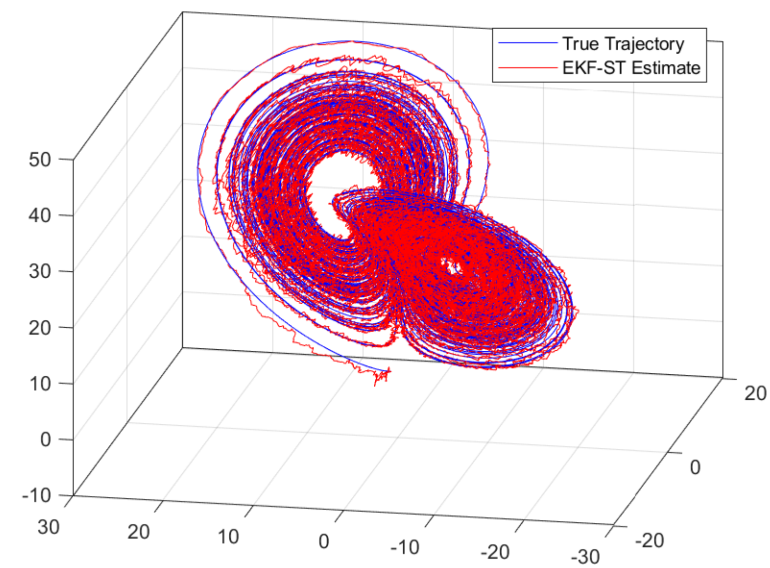

Figure 3: Experiment 2 true and estimated trajectories.

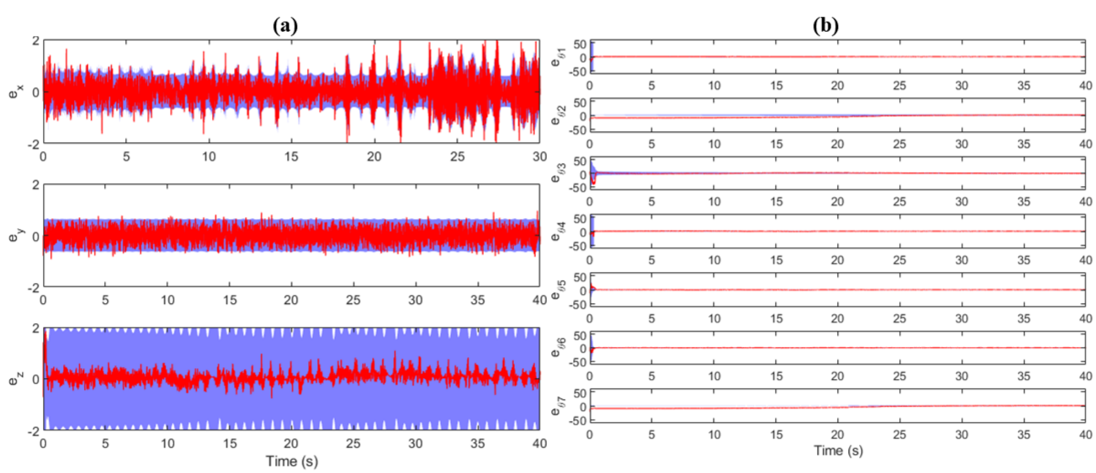

Figure 4: Experiment 2 error and 2-sigma bounds for (a) state estimations and (b) basis coefficient estimations.

The state estimation errors mostly stay within the 2-simgma bounds; there are inconsistencies in θ_2 and θ_7 estimates at first, but they eventually converge to the correct values.

3.3 Comparison with SINDy and EKF

Since the SINDy algorithm cannot handle partial state measurements, I let EKF-ST and EKF also have full state measurements in this test. For all three algorithms, the candidate basis used are 1, linear terms, and cross-terms. The true coefficients are then $\boldsymbol{\theta}_{\text{true}} = \begin{bmatrix} 0 & -10 & 10 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. The error of the equation learning results can be written as $\boldsymbol{e}_\theta = \boldsymbol{\theta}_{\text{true}} - \boldsymbol{\theta}_{\text{learned}}$. I tested the three algorithms with sensor noise standard deviations ranging from 1 to 5, and the error 2-norms are shown in Fig. 5

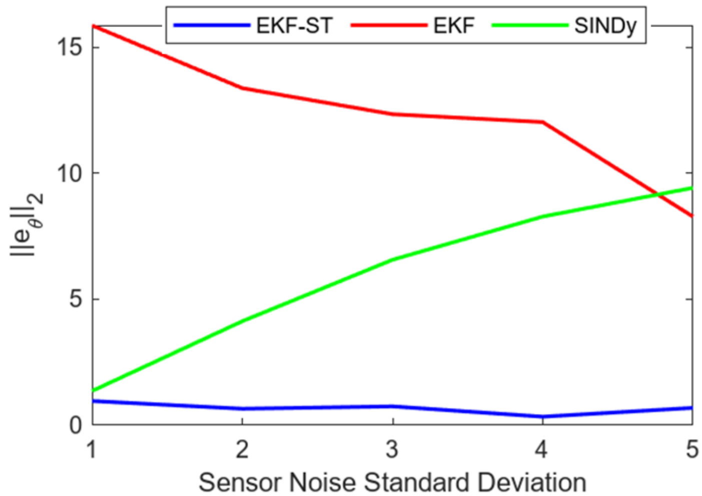

Figure 5: Equation learning error 2-norm comparisons.

In addition to smaller numerical error, the EKF-ST algorithm is also effective in avoiding spurious terms. At $\sigma_{\text{sensor}} = 3$, the equations learned via the three algorithms are

$$\dot{x}_{\text{EKF-ST}} = -10.5684x + 10.4792y$$
$$\dot{x}_{\text{EKF}} = 1.185 - 20.7732x + 15.8786y + 0.2736z + 0.0544xy - 0.247yz + 0.2661xz$$
$$\dot{x}_{\text{SINDy}} = 0.8242 - 5.0578x + 5.7614y$$

For reference, the true equation is

$$\dot{x} = -10x + 10y$$

4 Discussion

The numerical experiments show that EKF-ST provides consistent state estimations with noisy environments and measurements, full or partial state measurements, and partially known system dynamics. Moreover, the equation learning results from EKS-ST, particularly at high-noise regions, are more accurate than the results from ordinary EKF or SINDy, providing a smaller error norm and identifying the sparse structure of the true equation.

The key driver behind this improvement is the sequential thresholding stage: by actively zeroing coefficients whose posterior mean falls below λ_{ST}, the algorithm suppresses variance-driven false positives while still allowing coefficients to recover when evidence accumulates.

There are some limitations I found during the numerical experiments. First, the equation learning results from EKF-ST is very sensitive to the selection of λ_{ST} (more sensitive than the sparsity threshold in SINDy), and it is chanllenging to find a rigorous method of determining λ_{ST}. It is generally a good start to try $\lambda_{ST} = 0.1$, and adjust its value through multiple runs. Second, in this project, I only assume one equation is unkown; if more equations are unkown, then EKF-ST may be prone to obervability issues that are more significant than the standard EKF, since the sequential thresholding may set many of the elements in $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}_a}$ as 0, causing undesired sparsity in $\boldsymbol{H}_a \boldsymbol{F}_{k-1} \dots \boldsymbol{F}_1$.

5 Conclusions

This project proposes EKF-ST, a unified algorithm for simultaneous state estimation and online identification of sparse nonlinear dynamics. Numerical experiments show consistent state estimation results and accurate equation learning results that successfully identifies the true sparse pattern. The algorithm still has significant room for improvement, particularly in the selection of the threshold value, and a solution to the potential observability issue.

References

[1] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. "Discovering governing equations from data by sparse identification of nonlinear dynamical systems". In: *Proceedings of the national academy of sciences* 113.15 (2016), pp. 3932–3937.