Modeling and control of cyber-physical systems
Project Activity I

Sophie M. Fosson

April 4, 2025

In this project, we apply the mathematical models and algorithms discussed in class to estimate the state of a system, possibly in the presence of sensors attacks. In particular, we consider problems of target localization, in a two-dimensional indoor area.

The work is conceived for groups of 3-4 students. The choice of the programming language is free; we suggest MATLAB or Python.

Students are required to write a report ($\sim$ 4-5 pages) with the analysis of the obtained results.

Objectives

The goal of this activity is to learn to

1. implement algorithms for CPSs
2. enhance the algorithms to improve the performance (e.g., by a suitable tuning of the hyperparameters)
3. analyse the obtained results
4. write a technical report

Requirements

1. Implement the algorithms and solve the proposed problems
2. Write a report ($\sim$ 4-5 pages) that illustrates the analysis of the obtained results

3. Upload the report and the code in the delivery page of the course, at least one week before the oral examination

Task 1: Secure state estimation of a static CPS with sparse sensor attacks

- Consider P-Lasso to estimate the state of CPS under sparse sensors attacks, according to the model
$$y = C\widetilde{x} + \widetilde{a} + \eta$$
where $\widetilde{x} \in \mathbb{R}^n$ is the unknown state vector, $\widetilde{a} \in \mathbb{R}^q$ is the unknown sparse attack vector and $\eta \in \mathbb{R}^q$ a possible measurement noise. We aim at estimating the state and estimate which sensors are under attack

- Implement IJAM and ISTA to solve P-Lasso and compare their performance.

Suggested data and hyperparameters:

1. $n = 15$, $q = 30$, $h = 2$ sensor attacks
2. Generate the components of C according to a standard normal distribution $\sim \mathcal{N}(0, 1)$
3. Support $\mathbf{S}$ of the attack vector a, e.g., which sensors are under attack: generated randomly with uniform distribution
4. Attack: $\widetilde{a}_i \in [-5, -4] \cup [4, 5]$, for each $i \in \mathbf{S}$, generated randomly with uniform distribution
5. State: $\widetilde{x}_j \in [-3, -2] \cup [2, 3]$, for each $j = 1, \ldots, n$, generated randomly with uniform distribution
6. Measurement noise $\eta \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 10^{-2}$
7. Stop criterion: T_{max} = first step such that $\|x(T_{max}+1) - x(T_{max})\|_2^2 < \delta$, $\delta = 10^{-10}$.
8. $\lambda = 0.1$
9. For ISTA: $\nu = \frac{0.99}{\|G\|_2^2}$ where $G = \begin{pmatrix} C & I \end{pmatrix}$
10. For IJAM: $\nu = 0.7$

Repeat the experiment for at least 20 runs and analyse the mean results, by considering the following recovery performance metrics:

1. State estimation error, defined as $\frac{\|x(k)-\widetilde{x}\|_2}{\|\widetilde{x}\|_2}$, which measures the accuracy of the estimated state $x(k)$ with respect to the true state
2. Support attack error, calculated as $\sum_j |\mathbf{1}(\widetilde{a}_j \neq 0) - \mathbf{1}(a_j(k) \neq 0)|$, where $\mathbf{1}(v) = 1$ if v is true, and 0 otherwise, assessing the correctness of the attack support estimation.
3. The results should be averaged over the number of runs.

Questions:

1. Verify if ISTA and IJAM achieve the same recovery performance metrics
2. Analyze the convergence rate of ISTA and IJAM
3. Test several values of λ and comment the results
4. Test several values of ν and comment the results
5. Resilience to attacks: increase h and comment the results

Task 2: Target localization under sparse sensor attacks

We consider an indoor localization problem with an RSS fingerprinting setting.

The dictionary D and the run-time measurements y are given in file localization_data.mat.

We aim at localizing 1 target in 100 m^2 area, split into $n = 100$ cells. A sensor network with $q = 20$ sensors is randomly deployed in the room; see Fig. 4. Some sensors are tampered by adversarial attacks; we aim at identifying which sensors are under attack.

To localize the target and identify the sensors under attack, implement ISTA to solve the following weighted Lasso

$$\min_{x\in\mathbb{R}^n, a\in\mathbb{R}^q} \left\| G \begin{pmatrix} x \\ a \end{pmatrix} - y \right\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|a\|_1 \tag{1}$$

where $G = \text{normalize}\begin{pmatrix} D & I \end{pmatrix}$. In this way, the columns of G have mean $= 0$ and variance $= 1$; normalization is recommended to ensure that the columns of G are on the same scale.

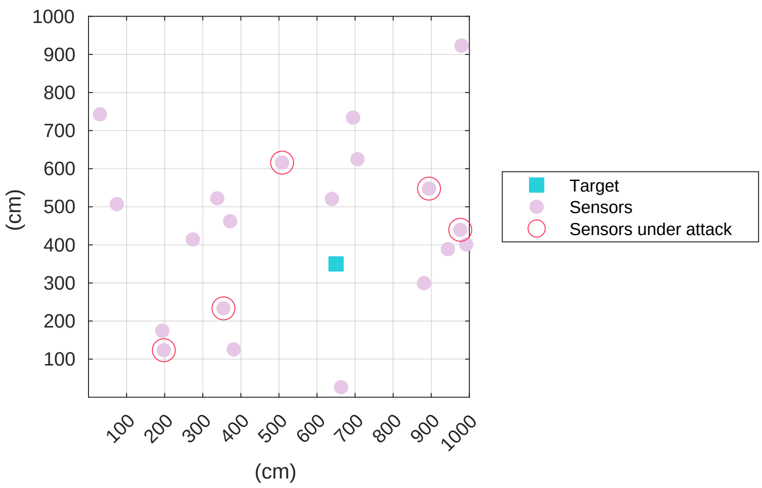

Figure 1: Localization problem

Suggested parameters:

1. $\lambda_1 = \lambda_2 = \lambda = 10$
2. $\nu = \|G\|_2^{-2}$

Algorithm 1 ISTA for problem (1)

1: Initialization: $x(0) = 0 \in \mathbb{R}^n$, $a(0) = 0 \in \mathbb{R}^q$
2: **for all** $k = 0, \ldots, T_{max}$ **do**
3: $$\begin{pmatrix} x(k+1) \\ a(k+1) \end{pmatrix} = \mathbb{S}_{\nu\lambda}\left[\begin{pmatrix} x(k) \\ a(k) \end{pmatrix} - \nu G^\top \left(G\begin{pmatrix} x(k) \\ a(k) \end{pmatrix} - y\right)\right]$$
4: **end for**

[Solution: supp$(\widetilde{x}) = \{37\}$, supp$(\widetilde{a}) = \{1, 10, 14, 16, 17\}$]

Task 3: Secure state estimation of a dynamic CPS with sparse sensor attacks

The data provided in the file dynamic_CPS_data.mat describe a dynamic CPS

$$x(k+1) = Ax(k)$$
$$y(k) = Cx(k) + a$$

with sparse, constant sensor attacks. The setting is as follows.

- Number of sensors: $q = 30$
- Number of sensors under attack: $h = 3$
- State dimension: $n = 15$

Goal: online tracking of the state and estimation of the attacks, by implementing a suitable observer.

The following points must be addressed.

1. Why the classic Luenberger observer is not a good strategy for the given CPS?
2. Implement SSO and D-SSO and verify that the attacks are identified at a finite time.
3. Analyse and compare the behaviors of SSO and D-SSO, by considering the following performance metrics:
   (a) State estimation error, defined as $\frac{\|\hat{x}(k)-x(k)\|_2}{\|x(k)\|_2}$
   (b) Support attack error, calculated as $\sum_j |\mathbf{1}(a_j \neq 0) - \mathbf{1}(\hat{a}_j(k) \neq 0)|$, where $\mathbf{1}(v) = 1$ if v is true, and 0 otherwise, assessing the correctness of the attack support estimation.

Suggested hyperparameters:

- $\lambda = 0.1$
- For SSO: $\nu = \frac{0.99}{\|G\|_2^2}$ where $G = \begin{pmatrix} C & I \end{pmatrix}$
- For D-SSO: $\nu = 0.7$

Extra questions (optional)

- In Fig. 2, we can see that the state estimation error is not null. In Fig. 3, we depict a refined solution. Think about possible policies to achieve refined solutions.

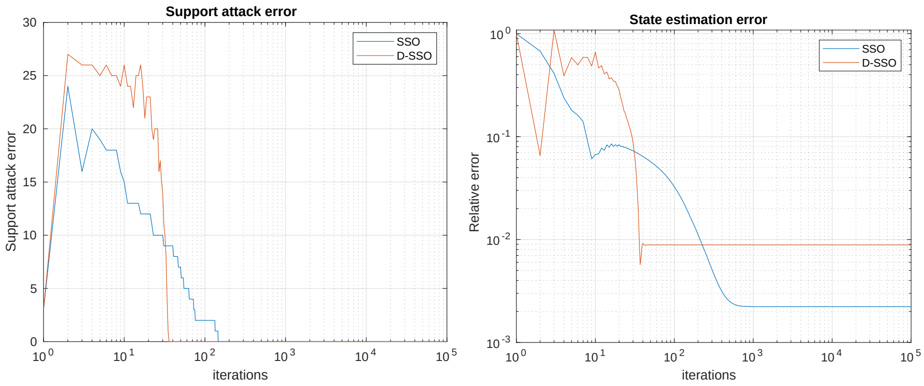

Figure 2: Task 3: Results

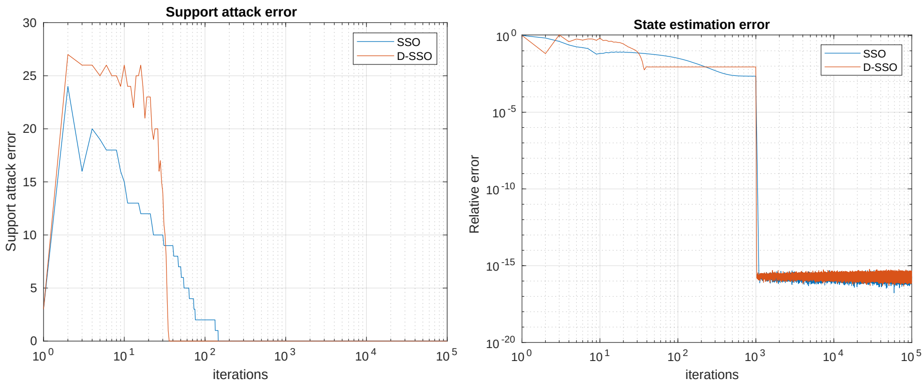

Figure 3: Task 3: Refined results

Task 4: Target tracking under sparse sensor attacks

We consider an indoor tracking problem with an RSS fingerprinting setting.

The model is

$$x(k+1) = Ax(k)$$
$$y(k) = Dx(k) + \widetilde{a} + \eta$$

where η is a measurement noise, for $k = 0, \ldots, T$ We provide D, $y(k)$, A in the file tracking_data.mat.

We also provide $\widetilde{a}$ (atrue) and $x(0)$ (xtrue0) for final analysis.

We aim at tracking 1 moving target in square are split into $n = 36$ cells. A sensor network with $q = 15$ sensors is randomly deployed in the room; see Fig. 4. Some sensors are tampered by adversarial attacks; we aim at identifying which sensors are under attack.

To track the target and identify the sensors under attack, implement SSO with the following modification: since the state is sparse, apply the soft thresholding also on the state update (see Task 2).

As in Task 2, $G =$ normalize$\begin{pmatrix} D & I \end{pmatrix}$. In this way, the columns of G have mean $= 0$ and variance $= 1$; normalization is recommended to ensure that the columns of G are on the same scale.

Suggested parameters:

1. $\lambda_1 = \lambda_2 = \lambda = 10$
2. $\nu = \|G\|_2^{-2}$

Analysise the results based on the following performance metrics

1. Support attack error, calculated as $\sum_j |\mathbf{1}(|\widetilde{a}_j| \geq \epsilon) - \mathbf{1}(|\hat{a}_j(k)| \geq \epsilon)|$, where $\mathbf{1}(v) = 1$ if v is true, and 0 otherwise, $\epsilon = 1$
2. Support state error, calculated as $\sum_j |\mathbf{1}(|\widetilde{x}_j(k)| \geq \epsilon) - \mathbf{1}(|\hat{x}_j(k)| \geq \epsilon)|$, where $\mathbf{1}(v) = 1$ if v is true, and 0 otherwise, $\epsilon = 1$

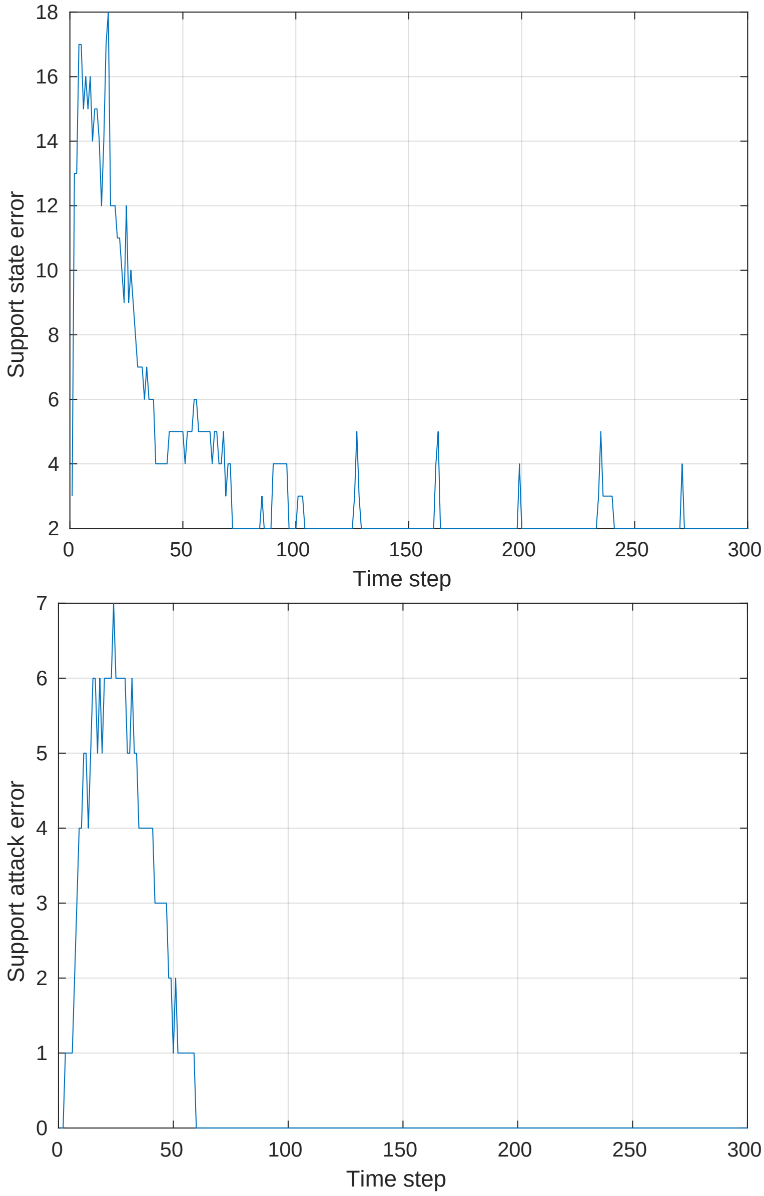

Figure 4: Tracking problem