

# Modeling and control of cyber-physical systems

## Project I

Sophie M. Fosson

April 12, 2024

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) with 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: Implementation of ISTA

---

Given  $C \in \mathbb{R}^{q,p}$ , a  $h$ -sparse  $\tilde{x} \in \mathbb{R}^p$ , and noisy measurements  $y = C\tilde{x} + \eta$ , where  $\eta \in \mathbb{R}^q$  is the measurement noise, the LASSO problem is:

$$\min_{x \in \mathbb{R}^p} \frac{1}{2} \|Cx - y\|_2^2 + \lambda \|x\|_1$$

where  $\lambda \in \mathbb{R}$  is a design hyperparameter.

Solve LASSO by implementing ISTA.

---

### Algorithm 1 ISTA

---

- 1: Initialization:  $x(0) = 0 \in \mathbb{R}^p$
  - 2: **for all**  $k = 0, \dots, T_{max}$  **do**
  - 3:    $x(k+1) = \mathbb{S}_{\tau\Lambda} [x(k) + \tau C^\top (y - Cx(k))]$
  - 4: **end for**
- 

Given  $x \in \mathbb{R}^p$ , the shrinkage/thresholding operator  $\mathbb{S}_\gamma : \mathbb{R}^p \mapsto \mathbb{R}^p$ , with  $\gamma = (\gamma_1, \dots, \gamma_p)^\top \in \mathbb{R}_+^p$ , is defined as

$$\mathbb{S}_{\gamma_i}(x_i) := \begin{cases} x_i - \gamma_i & \text{if } x_i > \gamma_i \\ x_i + \gamma_i & \text{if } x_i < -\gamma_i \\ 0 & \text{if } |x_i| \leq \gamma_i \end{cases}$$

for each  $i = 1, \dots, p$ .

Suggested setting:

1.  $\tau = \|C\|_2^{-2} - \epsilon$ ,  $\epsilon = 10^{-8}$
2.  $q = 10$ ,  $p = 20$ ,  $k = 2$ ,  $\lambda = \frac{1}{100\tau}$   
( $\Rightarrow \Lambda = \lambda(1, \dots, 1)^\top$ ,  $\tau\Lambda = 10^{-2}(1, 1, \dots, 1)^\top$ ).
3. Generate the components of  $C$  according to a standard normal distribution  $\sim \mathcal{N}(0, 1)$  (randn)
4. Generate the support  $S$  of the true  $\tilde{x}$  with uniform distribution and the non-zero components  $i \in S$  such that  $\tilde{x}_i \in [-2, -1] \cup [1, 2]$ , with uniform distribution
5. Measurement noise  $\eta \sim \mathcal{N}(0, \sigma^2)$ ,  $\sigma = 10^{-2}$  ( $\sigma * \text{randn}$ )

6. Stop criterion:  $T_{max}$  = first step such that  $\|x(T_{max} + 1) - x(T_{max})\|_2 < \delta$ ,  $\delta = 10^{-12}$ .

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

1. Support recovery rate: how many times the support of  $\tilde{x}$  is correctly estimated?
2. Can we obtain 100% of success in the support recovery by increasing  $q$ ?
3. Convergence time: how many iterations are required (mean, min, max)?
4. Try different values for  $\tau$ , by keeping  $\tau\lambda$  constant
5. Try different values for  $\lambda$ , by keeping  $\tau$  constant

## Task 2: Secure estimation under sparse sensor attacks

---

Reminder: we can extend LASSO by assigning a different  $\ell_1$  weight to each component, e.g.,

$$\min_{x \in \mathbb{R}^p} \frac{1}{2} \|Cx - y\|_2^2 + \sum_{i=1}^p \lambda_i |x_i|.$$

Reformulate the previous problem to estimate a (non-sparse)  $\tilde{x} \in \mathbb{R}^n$  under sparse sensor attacks, by using ISTA, given the measurements  $y = C\tilde{x} + a + \eta$ , where  $\eta \in \mathbb{R}^q$  is the measurement noise.

Suggested data and hyperparameters:

1.  $n = 10$ ,  $q = 20$ ,  $h = 2$  sensor attacks
2.  $C \sim \mathcal{N}(0, 1)$ ;  $\tilde{x} \sim \mathcal{N}(0, 1)$
3. Support of the attack vector  $a$ : uniform distribution
4. How to perform the attacks?
  - “Unaware” attack:  $a_i \in [-2, -1] \cup [1, 2]$ , uniformly distributed
  - “Aware” attack: the attacker takes the sensor measurement  $y_i$  and corrupts it of a quantity equal to  $\frac{1}{2}y_i$
5.  $\tau\Lambda = \tau\lambda(0, \dots, 0, 1, \dots, 1)^\top \in \mathbb{R}^{n+q}$ ,  $\tau\lambda = 2 \times 10^{-3}$
6. Measurement noise  $\eta = 0$  and  $\eta \sim \mathcal{N}(0, \sigma^2)$ ,  $\sigma = 10^{-2}$  ( $\sigma$  \*randn)

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

1. Rate of attack detection: how many times the support of  $a$  is correctly estimated, i.e., we identify the sensors under attack?
2. Estimation accuracy: is the estimation of  $\tilde{x}$  accurate? Compute  $\|\tilde{x} - \hat{x}\|_2^2$  where  $\hat{x} = x(T_{max})$  is the obtained estimate.

### Task 3: Target localization under sparse sensor attacks

---

We consider an indoor localization problem with an RSS fingerprinting setting. We propose a localization problem where 3 targets are deployed in a square room with  $p = 100$  cells. A sensor network with  $q = 25$  sensors is randomly deployed in the room, see the figure. The sensors measure the RSS from targets. We assume that *few* sensors are under attack. The goals are

1. Localize the targets, e.g., estimate which cells of the grid are occupied by a target
2. Find which sensors are under attack.

Optional task: compare the obtained results to the results of other methods (e.g.,  $k$ -NN), in terms of estimation accuracy and computational complexity/run time.

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

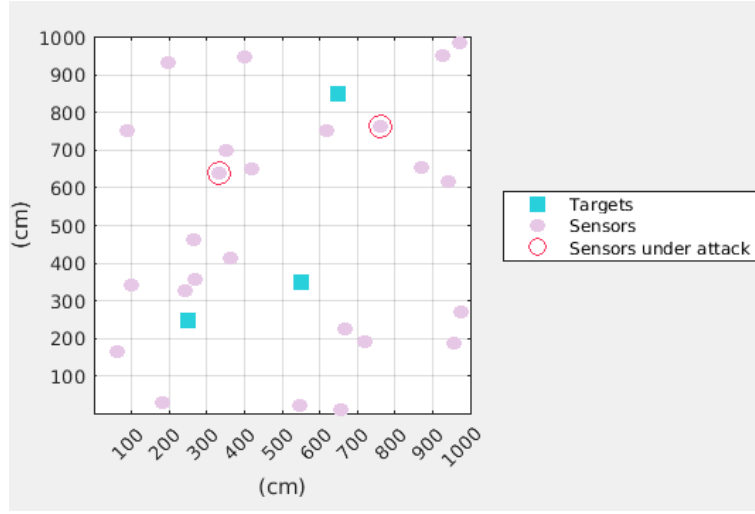
To localize the targets and identify the sensors under attack, implement ISTA to solve the following weighted Lasso

$$\min_{x \in \mathbb{R}^p, a \in \mathbb{R}^q} \left\| (D, I) \begin{pmatrix} x \\ a \end{pmatrix} - y \right\|_2^2 + \lambda_1 \|x\|_1 + \lambda_2 \|a\|_1$$

Hints

1.  $\lambda_1 = 10, \lambda_2 = 20$
2.  $G = [D, I] \rightarrow$  normalize  $G$ , e.g., by using `normalize(G)` in MATLAB
3.  $\tau = \|G\|_2^{-2} - \epsilon$

[Solution:  $\text{supp}(\tilde{x}) = \{23, 36, 87\}$ ,  $\text{supp}(\tilde{a}) = \{12, 16\}$ ]



## Task 4: Sparse observer

We consider the problem of Task 3 with moving targets and sparse sensor attacks. There are 3 targets, moving according to the dynamics

$$x(k+1) = Ax(k).$$

$A$  is given in the file `tracking_moving_targets.mat`. Basically, at each time step, the targets move towards left of one cell. The given  $Y \in \mathbb{R}^{q,50}$  are the measurements for 50 time instants, i.e., for  $k = 0, \dots, 49$ .

Implement the sparse observer to track the moving targets and to identify which sensors are under attack.

Notation:

$$z(k) = \begin{pmatrix} x(k) \\ a(k) \end{pmatrix} \quad \hat{z}(k) = \begin{pmatrix} \hat{x}(k) \\ \hat{a}(k) \end{pmatrix} \quad \hat{z}^+(k) = \begin{pmatrix} \hat{x}^+(k) \\ \hat{a}^+(k) \end{pmatrix} \quad G = \begin{pmatrix} D & I \end{pmatrix}$$

Hint:  $\lambda = (10, \dots, 10, 20, \dots, 20)$

---

### Algorithm 2 Sparse observer

---

- 1: **for all**  $k = 0, \dots, 49$  **do**
  - 2:    $\hat{z}^+(k) = \mathbb{S}_{\tau\lambda} [\hat{z}(k) + \tau G^\top (y(k) - G\hat{z}(k))]$
  - 3:    $\hat{x}(k+1) = A\hat{x}^+(k)$
  - 4:    $\hat{a}(k+1) = \hat{a}^+(k)$
  - 5: **end for**
- 

Analyse the following points:

1. Does the sparse observer converge?
2. After how many iterations the sparse observer converges?

Optional tasks:

1. Implement an “aware” time-varying attack: the attacker attacks two sensors by taking their measurements  $y_i(k)$  and by adding a quantity equal to  $a_i(k) = \frac{1}{2}y_i(k)$
2. Implement a time-varying attack where also the sensors under attack change in time; for example, two sensors are under attack for  $k = 0, \dots, 24$  and two other sensors are under attack for  $k = 25, \dots, 49$ .
3. Increase the number of sensors under attack. Which is the maximum number of attacks which allows us to perform a correct tracking of the targets?