MODELING AND CONTROL OF CYBER-PHYSICAL SYSTEMS

1 Modeling of Cyber-Physical Systems

Mathematical modeling for CPS

$$\text{CPSs} \xrightarrow{\text{set of devices that}} \begin{cases} \text{inter-communicate} \\ \text{compute} \\ \text{interact with the physical-world} \end{cases} \xRightarrow{\text{math models}} \begin{cases} \text{Hybrid systems (dynamic switching)} \\ \text{CPSs under attacks (\textbf{Part I})} \\ \text{Multi-agent systems (\textbf{Part II})} \end{cases}$$

$$\textbf{CPS under attacks (on the sensors)} \longrightarrow \begin{cases} x(k+1) = Ax(k) \\ y(k) = Cx(k) + a(k) \end{cases} \quad \text{no } u(k), \text{ no } b(k)\text{(actuators)}$$

$A \in \mathbb{R}^{n,n}$, $\quad C \in \mathbb{R}^{q,n} \quad a(k) \in \mathbb{R}^q$, $\quad q =$ number of sensors

Secure State Estimation of CPSs [FUSION CENTER]

Without attacks Observability $\iff rank(\mathcal{O}_n) = n$, $\mathcal{O}_n = [C \; CA \; ... \; CA^{n-1}]^T$ (observability matrix) $\Rightarrow x(0)$ found by pseudo-inversion of the system $y = \mathcal{O}_n x(0) \longrightarrow x(0) = \mathcal{O}_n^{\dagger} \, y$

Presence of attacks no model for the attacks $\Longrightarrow$ $\begin{cases} y(0) = Cx(0) + a(0) \\ y(1) = CAx(0) + a(1) \\ \vdots \qquad \vdots \\ y(T) = CA^{T-1}x(0) + a(T) \end{cases}$ (∞ solutions)

Assumption (sparsity) $\|a\|_0 \leq h \ll q$ (by adding this the problem may have a **unique solution**)

Under which conditions can we solve the problem? → Simplification of the problem

$A = I_n$ (the system is static) $\longrightarrow x(k+1) = x(k) = x \xrightarrow{\text{problem becomes}} y = Cx + a \quad \text{s.t. } \|a\|_0 \leq h$

Prop(Correctability) (resilience to h attacks) $\iff \forall z \in \mathbb{R}^n, \; \|\mathcal{O}_T z\|_0 > 2h$

A necessary condition for correctability... $q \geq 2h + n$ $\begin{cases} \text{large } q \text{ is not sufficient} \\ \text{a minimum } q \text{ is required} \end{cases}$

How can I solve the problem? → Reformulation of the problem

0. if **Correctable**, the Decoder $\mathcal{D}_0 \doteq \min_{x \in \mathbb{R}^n, a \in \mathbb{R}^q} \|a\|_0$ s.t. $y = Cx + a$ corrects h attacks;
1. $y = Cx + a +$ noise $\longrightarrow y \sim Cx + a \longrightarrow \min_{x \in \mathbb{R}^n} \frac{1}{2}\|y - Cx\|_2^2$ (Least-Squares problem)
2. ℓ_0-norm $\Rightarrow$ bad function $\underset{\text{convex relaxation}}{\Longrightarrow}$ ℓ_1-norm (best convex approximation) $\leftarrow$ **promotes sparsity**

$$(0)+(1)+(2) \Longrightarrow z = \underset{x \in \mathbb{R}^n, a \in \mathbb{R}^q}{\operatorname{argmin}} \tfrac{1}{2}\|y - Gz\|_2^2 + \lambda \|a\|_1 \quad G = (C \quad I), \; z = \begin{pmatrix} x \\ a \end{pmatrix} \Longleftarrow \begin{cases} \text{convex and continuous} \\ \textbf{unconstrained} \\ \text{non differentiable in 0} \end{cases}$$

(Partial) LASSO $\xrightarrow{\text{solved by using}}$ $x(k+1) = \mathbb{S}_{\lambda\tau}[x(k) + \tau C^T(y - Cx(k))]$ (**IST** algorithm), $k \to$ iteration

$\Longleftarrow$ derived from the **alternating minimization of the surrogate functional** $\mathcal{R}(x, b)$

In general for CPSs x is not sparse $\longrightarrow \Lambda = (0, 0, ..., 0, \lambda, \lambda, ..., \lambda)$, $\lambda_i = 0, \; i = 1, ..., n$

(Indoor) Localization by RSS-fingerprinting [FUSION CENTER]

0. **Init**: cell-grid discretization (p=# of cells, q=# of sensors, N_t=# of targets)
1. **Training phase**: Dictionary $D \in \mathbb{R}^{q,n}$ is built. Target in each cell + measurement
2. **Runtime phase**: each sensor takes $y_i \longrightarrow$ **Where are the targets?** $y = Dx + \eta, \; x_i \in \{0, 1\}$
   $x_i = 1 \to$ in the i-th cell there is a target $\Longrightarrow \min_{x \in \{0,1\}^p} \|y - Dx\|_2^2$ (mixed-integer $\to$ NP-Hard)
   (a) $x \in \{0, 1\}^p \underset{\text{relaxation}}{\longrightarrow} x \in \mathbb{R}^p$ so that the problem is feasible;
   (b) $N_t \ll p \Rightarrow$ seeking of a **sparse solution** $\Longrightarrow$ LASSO

(#1) Localization without attacks $x^* = \min_{x \in \mathbb{R}^p} \frac{1}{2}\|y - Dx\|_2^2 + \lambda \|x\|_1$ (ISTA can be used)

(#2) Localization under sensor attacks $x^* = \min_{z \in \mathbb{R}^p} \frac{1}{2}\|y - Gz\|_2^2 + \lambda \|z\|_1, \; G = (D \; I), \; z = (x \; a)^T$

Dynamic Secure State Estimation of CPSs [FUSION CENTER]

In general $A \neq I_n \Longrightarrow$ The system is moving and the static-batch approach is too slow

How to dynamically estimate the state (attack-free)?

Luemberger Observer (copy of the system + correction) $\underset{\text{def}}{\Longrightarrow} \begin{cases} \hat{x}(k+1) = A\hat{x}(k) - L[\hat{y}(k) - y(k)] \\ \hat{y}(k) = C\hat{x}(k) \end{cases} \underset{e(k)=\hat{x}-x}{\longrightarrow} e(k+1) = (A-LC)e(k)$

L is the **observer-gain matrix**, chosen such that $e(k+1)$ is asymptotically stable $\underset{k\to\infty}{\longrightarrow} \hat{x}(k) = x(k)$

Online Gradient Descent (OGD) $L_g = \tau A C^T, \quad \tau \leq \|C\|_2^{-2}$

What about the attacks?

In general CPSs under attacks are **not observable** $\longrightarrow$ NO Luemberger Observer, NO OGD

SPARSE OBSERVER $\underset{\text{given } \hat{z} \text{ and } y=Gz}{\overset{k \text{ is the time}}{\Longrightarrow}} \begin{cases} \hat{z}^+(k) = A\hat{z}(k) - \tau A G^T[G\hat{z}(k) - y(k)] & \text{estimation (OGD)} \\ \hat{x}(k+1) = A\hat{x}^+(k) & \text{prediction} \\ \hat{a}(k+1) = \mathbb{S}_{\lambda\tau}[\hat{a}^+(k)] & \text{sparsify the attacks} \\ \hat{y}(k) = G\hat{z}(k) & \text{measurement} \end{cases}$

Application: Localization under sensor attacks (with moving targets)

Distributed Secure State Estimation of CPSs [NO FUSION CENTER]

Multi-agent system set of systems which cooperate in order to achieve a **common goal**.

Agents $\underset{\text{exchange}}{\Longrightarrow}$ **local estimate** $x^{(i)}(k) \underset{\text{to the}}{\longrightarrow}$ neighbourhood

Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}) \rightarrow$ math tool to model the inter-communication. It is made up of:

- A set $\mathcal{N}$ of agents or nodes or vertices;
- A set $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ of links or edges

Neighbourhood $\mathcal{N}_i \rightarrow$ set of the nodes from which receives information.

Local mean $\bar{x}^{(i)}(k) = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} x^{(j)}(k)$

CONSENSUS $x(k+1) = Qx(k) \longrightarrow$ the system organized how Q dictates, achieves a **global common decision** if Q has got some properties.

Stochastic matrix Q is stochastic $\Longleftrightarrow \forall i = 1, ..., q \quad \sum_{j=1}^{q} Q_{ij} = 1$ **(row stochastic)**

Doubly stochastic matrix Q is doubly stochastic $\Longleftrightarrow$ is symmetric (undirected graphs)

Some important results:

$\longrightarrow \lambda_1 = \lambda_{PF} > |\lambda_2| > ... > |\lambda_q| \Longrightarrow \lim_{k\to\infty} x(k) = \alpha \mathbf{1}$ (suff. conditions for Consensus)

$\longrightarrow \lambda_{PF} > |\lambda_2| > ... > |\lambda_q|$, Q doubly stochastic $\underset{\text{average consensus}}{\Longrightarrow} \lim_{k\to\infty} x(k) = \alpha \mathbf{1}, \ \alpha = \frac{1}{q}\sum_{j=1}^{q} x^{(j)}(0)$

$\longrightarrow$ **Convergence rate** of the consensus algorithm $\underset{\text{is related to}}{\longrightarrow}$ $\texttt{esr}(\mathrm{Q})=\lambda_2$

$\longrightarrow$ If Q^T=Adjacency matrix describes a strongly connected graph $\Longrightarrow Q$ achieves consensus.

APPLICATIONS of CONSENSUS ALGORITHM

Distributed Least-Squares $\underset{\text{minimize}}{\longrightarrow} F(x) = \|y - Ax\|_2^2$, each sensor has $\begin{cases} y_i & \text{its own } \textbf{measurement} \\ A_i & i-\text{th row of the matrix A} \end{cases}$

$\Longrightarrow$ **Distributed Gradient Descent (DGD)** $\underset{\forall i \in \{1,...,q\}}{\longrightarrow} \begin{cases} \bar{x}^{(i)}(k) = \sum_{j=1}^{q} Q_{ij} x^{(j)}(k) \\ x^{(i)}(k+1) = \underbrace{\bar{x}^{(i)}(k)}_{\text{consensus step}} - \underbrace{\tau \nabla F(x^{(i)}(k))}_{\text{gradient step}} \end{cases}$

the solution can be found by using...

Distributed LASSO $\underset{\text{minimize}}{\longrightarrow} F(x) = \|y - Ax\|_2^2 + \lambda \|x\|_1$

$\Longrightarrow$ **Distributed ISTA(DISTA)** $\underset{\forall i \in \{1,...,q\}}{\longrightarrow} x^{(i)}(k+1) = \mathbb{S}_{\lambda\tau}\left[\bar{x}^{(i)}(k) + \tau A_i^T (y - A_i x^i(k))\right]$

the solution can be found by using...

Note that... for these algorithms proofs of convergence have been provided.
