MODELING AND CONTROL OF CYBER-PHYSICAL SYSTEMS

1 Modeling of Cyber-Physical Systems

Mathematical modeling for CPS

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

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

$A \in \mathbb{R}^{n,n}$, $C \in \mathbb{R}^{q,n}$ $a(k) \in \mathbb{R}^q$, $q =$ 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) $\|a\|_0 \le h \ll q$ (by adding this the problem may have a **unique solution**)

Under which conditions can we solve the problem? $\rightarrow$ Simplification of the problem

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

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

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

How can I solve the problem? $\rightarrow$ 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. $y = Cx + a + \text{noise} \longrightarrow y \sim Cx + a \longrightarrow \min_{x\in\mathbb{R}^n} \frac{1}{2}\|y - Cx\|_2^2$ (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) $\Longrightarrow z = \underset{x\in\mathbb{R}^n, a\in\mathbb{R}^q}{\text{argmin}} \frac{1}{2}\|y - Gz\|_2^2 + \lambda\|a\|_1$ $G = (C \quad I)$, $z = \begin{pmatrix} x \\ a \end{pmatrix}$ $\Longleftarrow \begin{cases} \text{convex and continuous} \\ \textbf{unconstrained} \\ \text{non differentiable in 0} \end{cases}$

(Partial) LASSO $\xrightarrow{\text{solved by using}} x(k+1) = \mathbb{S}_{\lambda\tau}[x(k) + \tau C^T(y - Cx(k))]$ (**IST** algorithm), $k \rightarrow$ 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?** $y = Dx + \eta$, $x_i \in \{0, 1\}$
   $x_i = 1 \rightarrow$ in the i-th cell there is a target $\Longrightarrow \min_{x\in\{0,1\}^p} \|y - Dx\|_2^2$ (mixed-integer $\rightarrow$ 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 $A \neq I_n \implies$ 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) $\underset{\text{def}}{\implies} \begin{cases} \hat{x}(k+1) = A\hat{x}(k) - L[\hat{y}(k) - y(k)] \\ \hat{y}(k) = C\hat{x}(k) \end{cases} \underset{e(k)=\hat{x}-x}{\longrightarrow} e(k+1) = (A-LC)e(k)$

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) $L_g = \tau AC^T, \quad \tau \leq \|C\|_2^{-2}$

What about the attacks?

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

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

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

Graph $\mathcal{G} = (\mathcal{N}, \mathcal{E}) \to$ 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 $\mathcal{N}_i \to$ set of the nodes from which receives information.

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

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

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

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

Some important results:

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

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

$\longrightarrow$ **Convergence rate** of the consensus algorithm $\underset{\text{is related to}}{\longrightarrow}$ esr(Q)=λ_2

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

<u>APPLICATIONS of CONSENSUS ALGORITHM</u>

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

the solution can be found by using...

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

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

the solution can be found by using...

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

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

2 Control of Cyber Physical Systems

Introduction

Leader node $S_0: \begin{cases} \dot{x}_0 = Ax_0 \\ y_0 = Cx_0 \end{cases}$ **Follower nodes** $S_i = \begin{cases} \dot{x}_i = Ax_i + Bu_i \\ y_i = Cx_i \end{cases} \quad i = 1, ..., N$

Communication network among the agents $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\},\ \mathcal{V} = \{v_1, v_2, ..., v_N\},\ \mathcal{E} \subset \mathcal{V} \times \mathcal{V}$

Augmented graph(Agents + Leader node) $\bar{\mathcal{G}} = \{\bar{\mathcal{V}}, \bar{\mathcal{E}}\},\ \bar{\mathcal{V}} = \{v_0, v_1, ..., v_N\},\ \bar{\mathcal{E}} = \bar{\mathcal{V}} \times \bar{\mathcal{V}}$

Agents' experimental modeling (System Identification)

Discrete time domain (LTI) state-space description $\begin{cases} \dot{x}_i(t) = Ax_i(t) + Bu_i(t) \\ y_i(t) = Cx_i(t) \end{cases}$

transfer function description $H(s) = C(sI - A)^{-1}B$

Continuous time domain (LTI) state-space description $\begin{cases} \dot{x}_i(k) = Ax_i(k) + Bu_i(k) \\ y_i(k) = Cx_i(k) \end{cases}$

transfer function description $H(z) = C(zI - A)^{-1}B$

Regression form $y(k) = f(y(k-1), y(k-2), ..., y(k-n), u(k), u(k-1), ..., u(k-m), \theta),\ m \leq n$

LEAST-SQUARES APPROACH

$$\underbrace{\begin{bmatrix} y(3) \\ y(4) \\ \vdots \\ y(H) \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} y(2) & y(1) & u(3) & u(2) & u(1) \\ y(3) & y(2) & u(4) & u(3) & u(2) \\ & \vdots & \vdots & \vdots & \vdots \\ y(H-1) & y(H-2) & u(H) & u(H-1) & u(H-2) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_H \end{bmatrix}}_{\theta}$$

Solution of LS: $\hat{\theta} = A^{\dagger}y = (A^TA)^{-1}A^Ty$ this is the solution of: $\hat{\theta} = \arg\min_{\theta} \|A\theta - y\|_2^2$

Least Squares properties $\lim_{H \to \infty} \mathbb{E}[\hat{\theta}] = \theta \iff$

(Consistency property)

- The error apper in the equation as an additive term $e(k)$ called the **equation error (EE)**
- The $e(k) k = 1, ..., H$ represents a white gaussian noise, that is the samples are *indipendent and identically distributed* (iid)

SET-MEMBERSHIP APPROACH [small amount of data + **mild assumption** on the noise]

Main ingredients

- $y(k) = f(y(k-1), ..., y(k-n), u(k), u(k-1), u(k-m), \theta_1, ..., \theta_{n+m+1}), \quad m \leq n$
- (1) A-priori assumption on the system m, n known + class of function $\mathcal{F}$ known
- A-priori **assumption on the noise**: Equation Error, Outuput Error, Error-In-Variable
- Input/Output noise are **bounded** $\rightarrow$ polynomial constraints $\iff$
- $\mathcal{B}_\eta = \{\eta: \ |\eta(k)| \leq \Delta_\eta\}, \quad \mathcal{B}_\xi = \{\xi: \ |\xi(k)| \leq \Delta_\xi\}$

Feasible Parameter Set (FPS)

$$\begin{aligned} \mathcal{D}_\theta = &\{\theta \in \mathbb{R}^p: \ y(k) = f(y(k-1), ..., y(k-n), u(k), u(k-1), ..., u(k-m), \theta),\ k = n+1, ..., H, \\ & y(k) = \tilde{y}(k) - \eta(k), \quad u(k) = \tilde{u}(k) - \xi(k),\ k = 1, ..., H | \xi(k)| \leq \Delta_\xi,\ |\eta(k)| \leq \Delta_\eta,\ k = 1, ..., H\} \end{aligned}$$

Extended Feasible Parameter Set (EFPS) (non convex -set defined by **polynomial constraints**)

$$\begin{aligned} \mathcal{D}_{\theta,\xi,\eta} = &\Big\{\theta \in \mathbb{R}^p,\ \xi \in \mathbb{R}^H,\ \eta \in \mathbb{R}^H: \ (\tilde{y}(k) - \eta(k)) + \sum_{i=1}^{n} \theta_i(\tilde{y}(k-1) - \eta(k-1)) = \\ & = \sum_{j=0}^{m} \theta_j(u(k-j) - \xi(k-j)),\ k = n+1, ..., H|\xi(k)| \leq \Delta_\xi,\ |\eta(k)| \leq \Delta_\eta,\ k = 1, ..., H\Big\} \end{aligned}$$

Parameter Uncertainty Intervals (PUIs) $\underline{\theta}_k = \min_{\theta,\xi,\eta \in \mathcal{D}_{\theta,\xi,\eta}} \theta_k, \quad \overline{\theta}_k = \max_{\theta,\xi,\eta \in \mathcal{D}_{\theta,\xi,\eta}} \theta_k, \; PUI_{\theta_k} = [\underline{\theta}_k, \overline{\theta}_k]$

Use of PUI
- In the correct form for **Robust control**: $\mathcal{H}_\infty$, μ-synthesis...
- In case we need a single model $\rightarrow$ **central estimate** defined as $\theta_k^c = \frac{\underline{\theta}_k + \overline{\theta}_k}{2}$
- **central estimate** $\iff$ Chebyshev center of $\mathcal{D}_\theta$ in ℓ_∞-norm

Other potentially availale a-priori info (can be encapsulated in EFPS)
- DC-GAIN: use of limits $s \rightarrow 0, \; z \rightarrow 1$ (linear constraints)
- BIBO stability of the system $\rightarrow$ enforced with Jury's Theorem

Convex relaxation for PUIs

Original optimization problem $\xrightarrow[\text{Moment Theory}]{}$ Set of SDPs $\xleftarrow[\text{depends on}]{}$ **Order of relaxation** δ It holds that:

$\lim_{\delta \to \infty} \underline{\theta}_k^\delta = \underline{\theta}_k, \quad \lim_{\delta \to \infty} \overline{\theta}_k^\delta = \overline{\theta}_k$ **Computational complexity**
- Exponential in δ
- Linear in number of parameters

`SparsePOP` data structures

Problem of type $\min_{x \in \mathbb{R}^n} \quad f_0(x) \quad \text{s.t.} f_k(x) \geq 0 \quad (k = 1, ..., l), f_k(x) = 0 \quad (k = l+1, ..., m), \texttt{lb}_i \leq x_i \leq \texttt{ub}_i$

Objective function f_0 (`objPoly`)
Constraints f_k (`ineqPolySys{k}`)
- `typeCone` $\rightarrow$ 1 (equality), -1 (inequality)
- `dimVar` $\rightarrow$ # of opt. variables including those in the constraints
- `degreee` $\rightarrow$ degree of f_0/f_k
- `noTerms` $\rightarrow$ number of monomials appearing in f_0/f_k
- `supports` $\rightarrow$ rows=noTerms, columns=dimVar
- `coef` $\rightarrow$ coefficients of the polynomial

Order of relaxation `param.relaxOrder`
Solution refinement `param.POPSolver='active-set'`

```
[param,SDPobjValue,POP,elapsedTime,SDPsolverInfo,SDPinfo]  = ...
    sparsePOP(objPoly,ineqPolySys,lbd,ubd,param);
```

Distributed optimal cooperative control of multi-agent systems (SVFB)

Cooperative tracking regulator $\iff$ The leader dictate a behaviour tracked by the agents

Neighbourhood of an agent $\mathcal{N}_i = \{j \mid a_{ij} > 0\}$ **Pinning matrix** $G = diag(g_1, g_2, ..., g_N)$
Neighbourhood tracking error $\varepsilon_i = \sum_{j=1}^N a_{ij}(x_j - x_i) + g_i(x_0 - x_i)$ **Local controller** $u_i = cK\varepsilon_i$
Closed-loop system dynamics $\dot{x}_i = Ax_i + Bu_i = Ax_i + cBK\left(\sum_{j=1}^N a_{ij}(x_j - x_i) + g_i(x_0 - x_i)\right)$
Laplacian matrix $L = [l_{ij}] = D - \mathcal{A}, \quad D = diag(d_1, d_2, ..., d_N)$

Global closed-loop dynamics $\dot{x} = (I_N \otimes A - c(L+G) \otimes BK)x + (c(L+G) \otimes BK)\underline{x}_0$

Disagreemnent error
- **Local** $\delta_i(t) = x_i(t) - x_0(t)$
- **Global** $\delta(t) = x(t) - \underline{x}_0(t) = col(\delta_1, \delta_2, \ldots, \delta_N)$.

Global disagreement error dynamics
$$\dot{\delta}_t = \dot{x}(t) - \dot{\underline{x}}_0 = A_c\delta(t),$$
$$A_c = I_N \otimes A - c(L+G) \otimes BK$$

Objective of cooperative tracking $\lim_{t \to \infty} \delta(t) = 0 \iff A_c$ is **Hurwitz**
Closed-loop eigenvalues $\text{eig}(A_c) = \bigcup_{i=1}^N \text{eig}(A - c\lambda_i BK)$ where λ_i are the eigenvalues of $L+G$

Cooperative controller design (K, c)
- $K = R^{-1}B^TP, \quad R$ properly selected **(controller gain matrix)**
- P solution of ARE: $A^TP + PA + Q - PBR^{-1}B^TP = 0$
- $Q, R \;\; \rightarrow u(t) = \underset{u(t)}{\text{argmin}}\left[\frac{1}{2}\int_0^{+\infty} \|x(t)\|_Q + \|u(t)\|_R \, dt\right]$ [opt. LQ]
- $c \geq \frac{1}{2\min_{i \in \mathcal{N}} Re(\lambda_i)}$ **(coupling gain)**

Dynamic of the leader S_0 $\quad \dot{x}_0(t) = Ax_0(t) - BK'x,\ K'$ in order to have $\begin{cases} \text{step} \to \text{one } \lambda_i = 0 \\ \text{ramp} \to \text{two coincident } \lambda_i \\ \text{sine} \to \text{complex conjugate } \lambda_i \end{cases}$

Dynamic regulator design for CPSs
GLOBAL OBSERVER DESIGN

Local output estimation error $\tilde{y}_i = y_i - \hat{y}_i = y_i - C\hat{x}_i$
Neighbourhood output estimation error $\xi_i = \sum_{j=1}^{N} a_{ij}(\tilde{y}_j - \tilde{y}_i) + g_i(\tilde{y}_0 - \tilde{y}_i)$
Local observer $\dot{\hat{x}}_i = A\hat{x}_i + Bu_i - cF\xi_i$

Global cooperative observer dynamics $\begin{matrix} \dot{\hat{x}} = A_o\hat{x} + (I_N \otimes B)u + c((L+G) \otimes F)y \\ A_o = (I_N \otimes A) - c((L+G) \otimes FC) \end{matrix}$

Global observer quantities $\begin{cases} \tilde{x}(t) = x(t) - \hat{x}(t) \to \text{ Global state estimation error} \\ \dot{\tilde{x}}(t) = \dot{x}_i(t) - \dot{\hat{x}}_i(t) = A_o\tilde{x}(t) \to \text{ estimation error dynamics} \\ \lim_{t\to\infty} \tilde{x}(t) = 0 \to \text{ Cooperative observer objective} \to A_o \textbf{ Hurwitz} \end{cases}$

Global Observer eigenvalues `to do...`

Global Observer design (F, c) `to do...`

COOPERATIVE DYNAMIC REGULATOR DESIGN `to do...`

Formation control

`to do...`