ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

Recovering Corrupting Errors---Analysis

Organizing the Observations for Further Processing—Final Form

- Let $\boldsymbol{\Omega}[k] = [\boldsymbol{I}_{\tau p} \quad \boldsymbol{F}[k]]$
- Let $\boldsymbol{E}[k] = [\boldsymbol{E}_s^\top[k] \quad \mathcal{V}^\top[k]]^\top$
- Then
$$\boldsymbol{Y}[k] = \mathcal{O}^{\tau-1}\boldsymbol{x}[k-\tau+1] + \boldsymbol{\Omega}[k]\boldsymbol{E}[k]$$
where $\boldsymbol{\Omega} \in \mathbb{R}^{\tau p \times [\tau p + (\tau-1)m]}$ and $\boldsymbol{E} \in \mathbb{R}^{\tau p + (\tau-1)m}$
- Objective: Recover $\boldsymbol{E}[k]$, a sparse vector

Computing Left Annihilator of $\mathcal{O}^{\tau-1}$

- Use MATLAB's `null` function to obtain

$$\boldsymbol{Q}_2^\top = \left(\operatorname{null}\left(\mathcal{O}^{\tau-1}\right)^\top\right)^\top,$$

where $\boldsymbol{Q}_2$ is right annihilator of $\left(\mathcal{O}^{\tau-1}\right)^\top$

- Indeed, since

$$\left(\mathcal{O}^{\tau-1}\right)^\top \boldsymbol{Q}_2 = \boldsymbol{O},$$

then taking the transpose gives

$$\boldsymbol{Q}_2^\top \mathcal{O}^{\tau-1} = \boldsymbol{O}^\top$$

Constructing an Optimization Problem for $\boldsymbol{E}[k]$ Recovery

- Recall that $\boldsymbol{Q}_2^\top \mathcal{O}^{\tau-1} = \boldsymbol{O}$
- Pre-multiply

$$\boldsymbol{Y}[k] = \mathcal{O}^{\tau-1}\boldsymbol{x}[k-\tau+1] + \boldsymbol{\Omega}[k]\boldsymbol{E}[k]$$

  by $\boldsymbol{Q}_2^\top$
- We obtain, $\boldsymbol{Q}_2^\top \boldsymbol{Y}[k] = \boldsymbol{Q}_2^\top \boldsymbol{\Omega}[k]\boldsymbol{E}[k]$
- Let $\boldsymbol{Z}[k] = \boldsymbol{Q}_2^\top \boldsymbol{Y}[k]$ and $\boldsymbol{W}[k] = \boldsymbol{Q}_2^\top \boldsymbol{\Omega}[k]$
- Then

$$\boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$$

The Constraint in the Optimization Problem to Recover $\boldsymbol{E}[k]$

- We have
$$\boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$$
where
$\boldsymbol{Z}[k] \in \mathbb{R}^{\tau p - n}$ and $\boldsymbol{W}[k] \in \mathbb{R}^{(\tau p - n) \times [\tau p + (\tau - 1)m]}$
- Note that $\boldsymbol{W}[k]$ is full row rank
- That is,
$$\operatorname{rank}(\boldsymbol{W}[k]) = \tau p - n$$
- This is because $\operatorname{rank}(\boldsymbol{Q}_2^\top) = \tau p - n$, $\operatorname{rank}(\boldsymbol{\Omega}[k]) = \tau p$
- Hence
$$\operatorname{rank}(\boldsymbol{W}[k]) = \operatorname{rank}(\boldsymbol{Q}_2^\top \boldsymbol{\Omega}[k]) = \operatorname{rank}(\boldsymbol{Q}_2^\top)$$

Optimization Problem to Recover $\boldsymbol{E}[k]$

If $\boldsymbol{E}[k]$ is an i-sparse vector, the solution to

$$\boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$$

can be obtained by solving the optimization problem

$$\min \|\boldsymbol{E}[k]\|_0 \quad \text{subject to} \quad \boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$$

Assumptions for the Optimization Problem to Recover $\boldsymbol{E}[k]$

- Assume that over the time interval $[k - \tau + 1, k]$ there are at most i_s malicious packet drops from the sensor to the controller and at most i_a malicious packet drops from the controller to the actuator
- Assume that $\boldsymbol{E}[k]$ is i-sparse
- Hence,

$$i = \|\boldsymbol{E}[k]\|_0 = \|\boldsymbol{E}_s[k]\|_0 + \|\boldsymbol{E}_a[k]\|_0 \leq i_s + i_a$$

Existence of the Solution to the Optimization Problem to Recover $\boldsymbol{E}[k]$

Lemma

If the solution $\boldsymbol{E}[k]$ *to* $\boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$ *is i-sparse and* $(\tau p - n) \geq 2(i_s + i_a)$ *and all subsets of* $2(i_s + i_a)$ *columns of* $\boldsymbol{W}[k]$ *are full rank, then* $\boldsymbol{E}[k]$ *is unique*

G. Fiore, Y. H. Chang, Q. Hu, M. D. Di Benedetto, C. J. Tomlin, *Secure state estimation for Cyber Physical Systems with sparse malicious packet drops*, 2017 ACC, Sheraton Seattle Hotel, Seattle, May 24–26, pp. 1898–1903

Approximating the Optimization Problem to Recover $\boldsymbol{E}[k]$

We approximate the the 0-norm optimization problem

$$\min \|\boldsymbol{E}[k]\|_0 \quad \text{subject to} \quad \boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$$

with the 1-norm optimization problem

$$\min \|\boldsymbol{E}[k]\|_1 \quad \text{subject to} \quad \boldsymbol{Z}[k] = \boldsymbol{W}[k]\boldsymbol{E}[k]$$

D. L. Donoho and M. Elad, *For most large under-determined systems of linear equations the minimal l_1-norm solution is also the sparsest solution*, SIAM Review, Vol. 56, No. 6, pp. 797–829, 2006

Converting the 1-Norm Optimization Into a Linear Programming Problem

- $\min \|\boldsymbol{E}[k]\|_1 = \min(\sum_{i=1}^{q} |E_i[k]|)$
- Let E_i^+, E_i^- be such that $|E_i| = E_i^+ + E_i^-$, $E_i = E_i^+ - E_i^-$, and $E_i^+ E_i^- = 0$
- Then we obtain

$$
\begin{array}{ll}
\min & (E_1^+ + E_1^-) + (E_2^+ + E_2^-) + \cdots + (E_q^+ + E_q^-) \\
\text{subject to} & \boldsymbol{W}(\boldsymbol{E}^+ - \boldsymbol{E}^-) = \boldsymbol{Z} \\
 & \boldsymbol{E}^+, \boldsymbol{E}^- \geq 0,
\end{array}
$$

where $\boldsymbol{E}^+ = [E_1^+ \quad \cdots \quad E_q^+]^\top$, $\boldsymbol{E}^- = [E_1^- \quad \cdots \quad E_q^-]^\top$, and $q = \tau p + (\tau - 1)m$

Linear Programming Program in Standard Form

- Let $\boldsymbol{x}_{lp} = [\boldsymbol{E}^{+\top} \quad \boldsymbol{E}^{-\top}]^\top$
- Let $\boldsymbol{c} = [1 \quad \cdots \quad 1]^\top \in \mathbb{R}^{2q}$
- Let $\boldsymbol{A}_{lp} = [\boldsymbol{W} \quad -\boldsymbol{W}]$
- Then we have

$$\begin{array}{ll} \min & \boldsymbol{c}^\top \boldsymbol{x}_{lp} \\ \text{subject to} & \boldsymbol{A}_{lp}\boldsymbol{x}_{lp} = \boldsymbol{Z} \\ \boldsymbol{x}_{lp} \geq 0 & \end{array}$$

- The above linear programming problem can be solved using standard methods

Recovering Output Sensor Error $\boldsymbol{e}_s[k]$

1. Choose τ such that $(\tau p - n) \geq 2(i_s + i_a)$
2. Compute $\boldsymbol{Y}[k]$ and matrices $\mathcal{O}^{\tau-1}$ and $\boldsymbol{\Omega}[k]$
3. Find left annihilator, $\boldsymbol{Q}_2^\top$ of $\mathcal{O}^{\tau-1}$
4. Construct the optimization problem, where $\boldsymbol{Z}[k] = \boldsymbol{Q}_2^\top \boldsymbol{Y}[k]$, $\boldsymbol{W}[k] = \boldsymbol{Q}_2^\top \boldsymbol{\Omega}[k]$
5. Solve optimization problem for $\tilde{\boldsymbol{E}}[k]$
6. Compute $\tilde{\boldsymbol{e}}_s[k]$ that approximates $\boldsymbol{e}_s[k]$

State Observer = UIO + Output Sensor Error Estimator