LESSON 11 - Recursive Least-Squares

OBJECTIVES: • INTRO. TO RECURSIVE FILTERS

CONSIDER ARX: $(m=n)$

$$\hat{y}[k] = -a_1 y[k-1] - \cdots - a_n y[k-n] + b_0 u[k] + \cdots + b_m u[k-n]$$

Define $\Phi_k = \begin{bmatrix} \phi_n \\ \vdots \\ \phi_k \end{bmatrix}_{(k-n)\times(2n+1)}$ where $\phi_{n+1} = [\underbrace{-y[n] \cdots y[1]}_{n} \ \underbrace{u[n+1] \cdots u[1]}_{n+1}]$

$\phi_k = [-y[k-1] \cdots y[0] \ u[k] \cdots u[k-n]]$ $(m=n)$

for $l \geq 3n$: $\underbrace{\left(\Phi_l^T \Phi_l\right)}_{(2n+1)(2n+1)} = \sum_{i=n}^{l} \phi_i^T \phi_i$

$l-n+1 \geq 2n+1$
$l \geq 3n$

P_k = Data (Inverse) Matrix at step k

$$P_k \triangleq \left(\sum_{i=n+1}^{k} \phi_i^T \phi_i\right)^{-1} = \left(\Phi_k^T \Phi_k\right)^{-1}_{(2n+1)\times(2n+1)}$$

Need to initialize: - Batch Least-squares until k_0: $\left(\Phi_{k_0}^T \Phi_{k_0}\right)^{-1}$

$$\therefore P_{k+1}^{-1} = \sum_{i=n+1}^{k+1} \phi_i^T \phi_i$$

$$= \sum_{i=n+1}^{k} \phi_i^T \phi_i + \phi_{k+1}^T \phi_{k+1}$$

$$= P_k^{-1} + \phi_{k+1}^T \phi_{k+1}$$

$$P_{k+1} = \left[P_k^{-1} + \phi_{k+1}^T \phi_{k+1}\right]^{-1}$$

by matrix inversion lemma

① $$\boxed{P_{k+1} = P_k - \frac{1}{1 + \phi_{k+1} P_k \phi_{k+1}^T} P_k \phi_{k+1}^T \phi_{k+1} P_k}$$

Updating a matrix inverse with calculating a matrix inverse.
(Computationally efficient.)

PARAMETER UPDATE:

$$\hat{\Theta}_k \triangleq \left(\Phi_k^T \Phi_k\right)^{-1} \Phi_k^T y_k$$ Least-squares soln at step k

↳ To initialize filter at $k=k_0$: $\hat{\Theta}_{k_0} = \left(\Phi_{k_0}^T \Phi_{k_0}\right)^{-1} \Phi_{k_0}^T y_{k_0}$ (Batch least-squares)

$$\hat{\Theta}_k = P_k \left[\phi_n^T \cdots \phi_{k-1}^T\right] \begin{bmatrix} y_n \\ \vdots \\ y_{k-1} \end{bmatrix}$$

$$= P_k \sum_{i=1}^{k-1} y_i \phi_i^T$$

$$= P_k \left(\sum_{i=n+1}^{k-1} y_i \phi_i^T + y[k]\phi_k^T \right)$$

$y_k = \phi_{k-1} \hat{\Theta}_{k-1}$

$y_{k+1} = \phi_k \hat{\Theta}_k$

$$= P_k \left[P_{k-1}^{-1} \hat{\Theta}_{k-1} + y[k]\phi_k^T \right]$$

$$= P_k P_{k-1}^{-1} \hat{\Theta}_{k-1} + P_k y[k] \phi_k^T$$

P_k^{-1}

$$\hookrightarrow P_k P_{k-1}^{-1} = P_k \left[\overbrace{\sum_{i=n+1}^{k-1} \phi_i^T \phi_i + \phi_k^T \phi_k}^{P_k^{-1}} - \phi_k^T \phi_k \right]$$

$$= I - P_k \phi_k^T \phi_k$$

$$(2) \quad \boxed{\hat{\Theta}_k = \hat{\Theta}_{k-1} + P_k \phi_k^T \left[y[k] - \phi_k \hat{\Theta}_{k-1} \right]}$$

→ filter alternates between (1) & (2)

$$\underbrace{\hat{\Theta}_k}_{\text{a posteriori estimate of parameters}} = \underbrace{\hat{\Theta}_{k+1}}_{\text{a priori estimate of parameters}} + \underbrace{P_k \phi_k^T}_{\text{gain}} \underbrace{\left[\underbrace{y[k]}_{\text{measurement of } y_k} - \underbrace{\phi_k \hat{\Theta}_{k-1}}_{\text{estimate of } y_k} \right]}_{\text{estimate error (aka residual)}}$$

Algorithm:
1) Initialize P_{k_0} & $\hat{\Theta}_{k_0}$ (Small Batch Least-Squares)
2) Sample new data point at step k
   ↳ $y[k]$, ($u[k]$ if measured)
3) update ϕ_k
4) update P_k using (1)
5) update $\hat{\Theta}_k$ using (2)
6) repeat steps 2–5 with remaining data.