# RECURSIVE IDENTIFICATION

## ARX IDENTIFICATION OUTLINE

- Least square } RECAP
- Recursive Least square (I, II, III) form
- Recursive Least square with forgetting factor

} NEW

## ARX System

$$y(t) = \frac{B(z)}{A(z)} u(t-1) + \frac{1}{A(z)} e(t) \qquad e(t) \sim WN(\mu_e, \lambda_e^2)$$

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_p z^{-p}$$

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_m z^{-m}$$

$$y(t) = -a_1 y(t-1) - \dots - a_m y(t-m) + b_0 u(t-1) + \dots + b_p u(t-p-1) + e(t)$$

$$\boxed{\vartheta = \begin{bmatrix} a_1 \\ \vdots \\ a_m \\ b_0 \\ \vdots \\ b_p \end{bmatrix} \qquad \varphi(t) = \begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-m) \\ u(t-1) \\ \vdots \\ u(t-p-1) \end{bmatrix} \implies y(t) = \varphi(t)^T \vartheta + e(t)}$$

objective: Identify $\vartheta$ starting from an available dataset

$$\{u(1) \dots u(N)\}$$
$$\{y(1) \dots y(N)\}$$

## LEAST SQUARE

$$\hat{\vartheta}_N = \underset{\vartheta}{\arg\min} \left\{ J_\vartheta(N) = \frac{1}{N} \sum_{t=1}^{N} \left( y(t) - \hat{y}(t|t-1, \vartheta) \right)^2 \right\} \longrightarrow \text{Prediction Error Method (P.E.M.)}$$

model predictor $\hat{y}(t|t-1)$

$$y(t) = \underbrace{\varphi(t)^T \vartheta}_{\text{totally known at time } t-1} + \underbrace{e(t)}_{\text{totally unpredictable at time } t-1}$$

$\hat{y}(t|t-1) = \varphi(t)^T \vartheta$ is the optimal 1-step predictor

$$J_N(\vartheta) = \frac{1}{N} \sum_{t=1}^{N} \left( y(t) - \varphi(t)^T \vartheta \right)^2 \leftarrow \text{quadratic function of } \vartheta$$

$\downarrow$

It is possible to find explicitely the minimizer

$\hat{\vartheta}$ is such that $\left.\dfrac{\partial J_N(\vartheta)}{\partial \vartheta}\right|_{\vartheta=\hat{\vartheta}_N} = 0$

$$\boxed{\begin{aligned}\hat{\vartheta}_N &= S(N)^{-1} \sum_{t=1}^{N} \varphi(t)\, y(t) \\ S(N) &= \sum_{t=1}^{N} \varphi(t)\, \varphi(t)^T\end{aligned}} \quad LS$$

DRAWBACK

$$\begin{matrix}\{u(1) \cdots u(N)\} \\ \{y(1) \cdots y(N)\} \\ \downarrow \\ \hat{\vartheta}_N\end{matrix} \xrightarrow{\text{New Measure}} \begin{matrix}\{u(1) \cdots u(N), u(N+1)\} \\ \{y(1) \cdots y(N), y(N+1)\} \\ \downarrow \\ \hat{\vartheta}_{N+1}\end{matrix}$$

$\hat{\vartheta}_N$ is not used to compute $\hat{\vartheta}_{N+1}$, it is necessary to repeat all the computation

SOLUTION $\longrightarrow$ RECURSIVE LEAST SQUARE

## RECURSIVE LEAST SQUARE

The recursive least square is a method that UPDATE the identification


ADVANTAGES

- Lower computational effort
- You save memory allocation

## Recursive Least square I-form

obj: $\hat{\vartheta}_N = f(\hat{\vartheta}_{N-1}, \varphi(N))$

$$\hat{\vartheta}_N = S(N)^{-1} \sum_{t=1}^{N} \varphi(t)\, y(t) \qquad (1)$$

from (1) $\displaystyle\sum_{t=1}^{N} \varphi(t)\, y(t) = S(N)\, \hat{\vartheta}_N \qquad (2)$

$$\hat{\vartheta}_{N-1} = S(N-1)^{-1} \sum_{t=1}^{N-1} \varphi(t)\, y(t) \qquad (3)$$

from (3) $\displaystyle\sum_{t=1}^{N-1} \varphi(t)\, y(t) = S(N-1)\, \hat{\vartheta}_{N-1} \qquad (4)$

$$\sum_{t=1}^{N} \varphi(t)\, y(t) = \sum_{t=1}^{N-1} \varphi(t)\, y(t) + \varphi(N)\, y(N) \qquad (5)$$

Replace (4) in (5)

$$\sum_{t=1}^{N} \varphi(t)\, y(t) = S(N-1)\, \hat{\vartheta}_{N-1} + \varphi(N)\, y(N) \qquad (6)$$

$(2) = (6)$
$$S(N)\, \hat{\vartheta}_N = S(N-1)\, \hat{\vartheta}_{N-1} + \varphi(N)\, y(N) \qquad (7)$$

Find an expression of $S(N-1)$

$$S(N) = \sum_{t=1}^{N} \varphi(t)\varphi(t)^T = \underbrace{\sum_{t=1}^{N-1} \varphi(t)\varphi(t)^T}_{S(N-1)} + \varphi(N)\varphi(N)^T$$

$$S(N-1) = S(N) - \varphi(N)\varphi(N)^T \qquad (8)$$

Replace (8) in (7)

$$S(N)\, \hat{\vartheta}_N = \left(S(N) - \varphi(N)\varphi(N)^T\right) \hat{\vartheta}_{N-1} + \varphi(N)\, y(N)$$

$$S(N)\, \hat{\vartheta}_N = S(N)\, \hat{\vartheta}_{N-1} - \varphi(N)\varphi(N)^T \hat{\vartheta}_{N-1} + \varphi(N)\, y(N)$$

$$\hat{\vartheta}_N = \hat{\vartheta}_{N-1} + \underbrace{S(N)^{-1}\varphi(N)}_{\substack{K(N)\\ \text{(gain)}}} \Big[ \underbrace{y(N) - \varphi(N)^T \hat{\vartheta}_{N-1}}_{\substack{\varepsilon(N)\\ \text{Prediction error}}} \Big]$$

$$\boxed{\begin{aligned} \hat{\vartheta}_N &= \hat{\vartheta}_{N-1} + K(N)\,\varepsilon(N) \\ K(N) &= S(N)^{-1}\varphi(N) \\ \varepsilon(N) &= y(N) - \varphi(N)^T \hat{\vartheta}_{N-1} \\ S(N) &= S(N-1) + \varphi(N)\varphi(N)^T \end{aligned}}$$

RLS - I

Drawback

$$S(N) - S(N-1) = \varphi(N)\varphi(N)^T \geq 0 \implies S(N) \xrightarrow[N \to \infty]{} \infty$$

$S(N)$ is a matrix wich continuously increases, and tends to saturate the numerical precision of the digital computing unit.

SOLUTION $\longrightarrow$ RECURSIVE LEAST SQUARE II

## Recursive Least square II form

Trick: normalization with respect to $N$

$$S(N) = S(N-1) + \varphi(N)\varphi(N)^T$$

$$R(N) = \frac{1}{N} S(N) \longrightarrow R(N) = \frac{1}{N} \underbrace{\frac{N-1}{N-1} S(N-1)}_{R(N-1)} + \frac{1}{N}\varphi(N)\varphi(N)^T$$

$$R(N) = \frac{N-1}{N} R(N-1) + \frac{1}{N}\varphi(N)\varphi(N)^T$$

$$\hat{\vartheta}_N = \hat{\vartheta}_{N-1} + K(N)\,\varepsilon(N)$$
$$K(N) = \frac{1}{N} R(N)^{-1} \varphi(N)$$
$$\varepsilon(N) = y(N) - \varphi(N)^T \hat{\vartheta}_{N-1}$$
$$R(N) = \frac{N-1}{N} R(N-1) + \frac{1}{N} \varphi(N)\varphi(N)^T$$

RLS - II

Drawback

at each time step a matrix inversion is required which requires a big computational effort

SOLUTION → RLS III

## Recursive least square III - form

LEMMA OF MATRIX INVERSION

consider 4 matrices F, G, H, K such that

- F + GHK makes sense (suitable dimensions)
- F, H and F+GHK are square invertible matrices

then

$$\underbrace{(F+GHK)^{-1}} = F^{-1} - F^{-1} G \left( H^{-1} + K F^{-1} G \right)^{-1} K F^{-1}$$

in our case $S(N)^{-1} = \left[ \underbrace{S(N-1)}_{F} + \underbrace{\varphi(N)}_{G} \underbrace{1}_{H} \underbrace{\varphi(N)^T}_{K} \right]^{-1}$

$$S(N)^{-1} = S(N-1)^{-1} - S(N-1)^{-1}\varphi(N) \underbrace{\left[ 1 + \varphi(N)^T S(N-1)^{-1} \varphi(N) \right]}_{\text{is a scalar}}^{-1} \varphi(N)^T S(N-1)^{-1}$$

define $V(N) = S(N)^{-1}$

$$V(N) = V(N-1) - \frac{V(N-1)\,\varphi(N)\,\varphi(N)^T\, V(N-1)}{1 + \varphi(N)^T V(N-1)\,\varphi(N)}$$

- V(N) does not tend to zero
- It is no more necessary to invert a matrix, instead a scalar inversion is required.

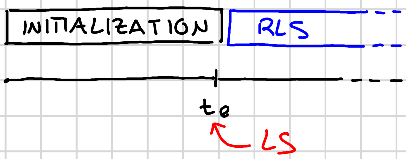
$$\hat{\vartheta}_N = \hat{\vartheta}_{N-1} + K(N)\,\varepsilon(N)$$
$$\varepsilon(N) = y(N) - \varphi(N)^T \hat{\vartheta}_{N-1}$$
$$K(N) = V(N)\,\varphi(N)$$
$$V(N) = V(N-1) - \frac{V(N-1)\varphi(N)\varphi(N)^T V(N-1)}{1 + \varphi(N)^T V(N-1)\varphi(N)}$$

RLS - III

Remark on initialization

RLS is a rigorous version of LS (not an approx) provided a correct initialization

CORRECT INITIALIZATION:



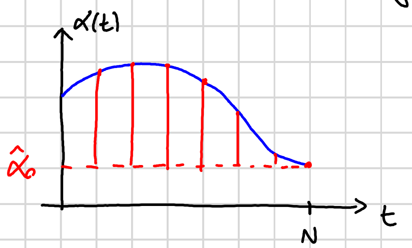

- Collect a first set of data till $t_0$
- Compute $\hat{\vartheta}_{t_0}$ and $S(t_0)$ with LS
- Use RLS to update $\vartheta$ and $S$

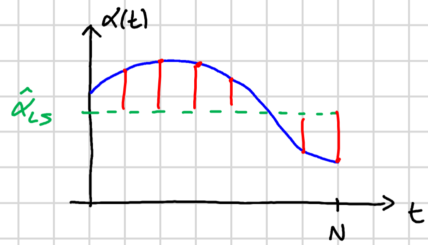
IN PRACTICE

$\hat{\vartheta}_0 = 0 \qquad S(0) = I$ $\longrightarrow$ the error introduced by this wrong initialization tends to vanish for $N \to \infty$

## Recursive Least square with forgetting factor

Consider a time varying parameter




$\hat{\alpha}_0$ is the correct estimation at time N, but it does not minimizes

$$J_N(\alpha) = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t|t-1,\alpha)\right)^2$$

because it consider the whole time History of the system

IN ORDER TO IDENTIFY A TIME VARYING PARAMETER THE RLS MUST BE FORCED TO FORGET OLD DATA

The solution is provided by the minimization of $J_N$

$$J_N(\vartheta) = \frac{1}{N}\sum_{t=1}^{N} \rho^{N-t}\left(y(t) - \hat{y}(t|t-1,\vartheta)\right)^2$$

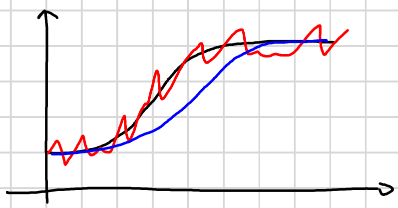
$0 \leq \rho \leq 1$ is called forgetting factor

$\rho = 1 \rightarrow$ previous simulation
$\rho < 1 \rightarrow$ the old data have less importance with respect to new data.

$$\boxed{\begin{aligned} \hat{\vartheta}_N &= \hat{\vartheta}_{N-1} + K(N)\,\varepsilon(N) \\ K(N) &= S(N)^{-1}\varphi(N) \\ \varepsilon(N) &= y(N) - \varphi(N)\,\hat{\vartheta}_{N-1} \\ S(N) &= \rho\, S(N-1) + \varphi(N)\,\varphi(N)^T \end{aligned}}$$

RLS with forgetting factor

Remark on choice of the forgetting factor



- $\rho \ll 1$ < high tracking speed / low precision
- $\rho \rightarrow 1$ $(\rho = 0.95)$ < low tracking speed / greater precision.