

## Recursive (On-line) Identification Methods

- Recursive Least Squares (RLS) Methods
- Forgetting Factor and Tracking Time-Varying Parameters
- Identification condition problem
- Computational Aspects

## Why?

Why is recursive identification of interest?

- On-line Estimation.
- Adaptive Systems.
- Time Varying Parameters.
- Fault Detection and Diagnosis.
- Simulation.

## How?

How do we estimate time-varying parameters?

- Update the model regularly (once every sampling instant)
- Make use of previous calculations in an efficient manner.
- The basic procedure is to modify the corresponding off-line method, *e.g.*, the block/batch least squares method, the prediction error method.

## Desirable Properties

We desire our recursive algorithms to have the following properties:

- Fast convergence.
- Consistent estimates (time-invariant models).
- Good tracking (for time-varying parameters, *e.g.* in the event of fault occurrence or operating condition changes).
- Computationally simple (perform all calculations during one sampling interval).

## Trade-offs

No algorithm is perfect. The design is always based on trade-offs, such as:

- Convergence versus tracking.
- Computational complexity versus accuracy.

## Recursive Least Squares Method (RLS)

$$\hat{\boldsymbol{\theta}}(t) = \arg \min_{\boldsymbol{\theta}} V_t(\boldsymbol{\theta}), \quad V_t(\boldsymbol{\theta}) = \sum_{k=1}^t \varepsilon^2(k)$$

where  $\varepsilon(k) = y(k) - \boldsymbol{\varphi}^T(k)\boldsymbol{\theta}$ . The solution reads:

$$\hat{\boldsymbol{\theta}}(t) = \mathbf{R}_t^{-1} \mathbf{r}_t$$

where

$$\mathbf{R}_t = \sum_{k=1}^t \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k), \quad \mathbf{r}_t = \sum_{k=1}^t \boldsymbol{\varphi}(k)y(k)$$

- The criterion function  $V_t(\boldsymbol{\theta})$  changes every time step, hence the estimate  $\hat{\boldsymbol{\theta}}(t)$  changes every time step.
- How can we find a recursive implementation of  $\hat{\boldsymbol{\theta}}(t)$ ?

### RLS

#### Algorithm:

At time  $t = 0$ : Choose initial values of  $\hat{\boldsymbol{\theta}}(0)$  and  $\mathbf{P}(0)$

At each sampling instant, update  $\boldsymbol{\varphi}(t)$  and compute

$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)\boldsymbol{\varepsilon}(t)$$

$$\boldsymbol{\varepsilon}(t) = y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}(t-1)$$

$$\mathbf{K}(t) = \mathbf{P}(t)\boldsymbol{\varphi}(t)$$

$$\mathbf{P}(t) = \left[ \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)}{1 + \boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} \right]$$

**Question:** How to obtain/derive this recursive version of LS from the block/batch LS?

## Tracking

How do we handle time-varying parameters? — two ways:

- Postulate a time-varying model for the parameters. Typically we let the parameters vary according to a random walk and use the Kalman filter as an estimator.
- Modify the cost function so that we gradually forget old data. Hence, the model is fitted to the most recent data (the parameters are adapted to describe the newest data).

- Modified cost function:

$$\hat{\boldsymbol{\theta}}(t) = \arg \min_{\boldsymbol{\theta}} V_t(\boldsymbol{\theta}), \quad V_t(\boldsymbol{\theta}) = \sum_{k=1}^t \beta(t, k) \varepsilon^2(k)$$

- Suppose that the weighting function  $\beta(t, k)$  satisfies

$$\begin{aligned} \beta(t, k) &= \lambda(t) \beta(t-1, k), \quad 0 \leq k < t \\ \beta(t, t) &= 1 \end{aligned}$$

A common choice is to let  $\lambda(t) = \lambda$ , where  $\lambda$  is referred to as a so-called *forgetting factor*. In this case we get:

$$\beta(t, k) = \lambda^{t-k}, \quad 0 < \lambda \leq 1$$

- $\lambda = 1$  corresponds to the standard RLS.

## Weighted RLS

### Algorithm:

At time  $t = 0$ : Choose initial values of  $\hat{\boldsymbol{\theta}}(0)$  and  $\mathbf{P}(0)$

At each sampling instant, update  $\boldsymbol{\varphi}(t)$  and compute

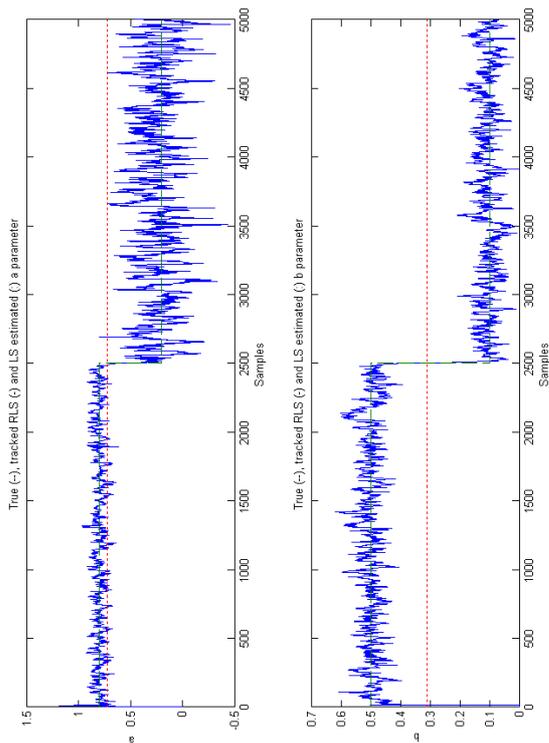
$$\hat{\boldsymbol{\theta}}(t) = \hat{\boldsymbol{\theta}}(t-1) + \mathbf{K}(t)\boldsymbol{\varepsilon}(t)$$

$$\boldsymbol{\varepsilon}(t) = y(t) - \boldsymbol{\varphi}^T(t)\hat{\boldsymbol{\theta}}(t-1)$$

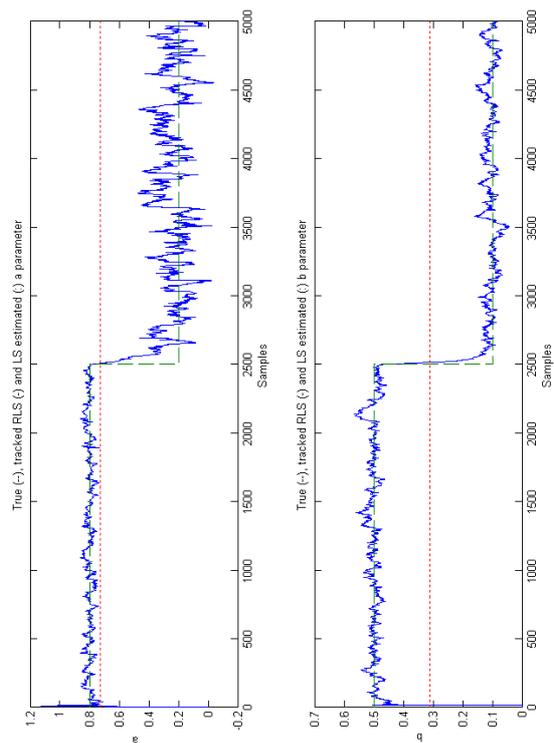
$$\mathbf{K}(t) = \mathbf{P}(t)\boldsymbol{\varphi}(t)$$

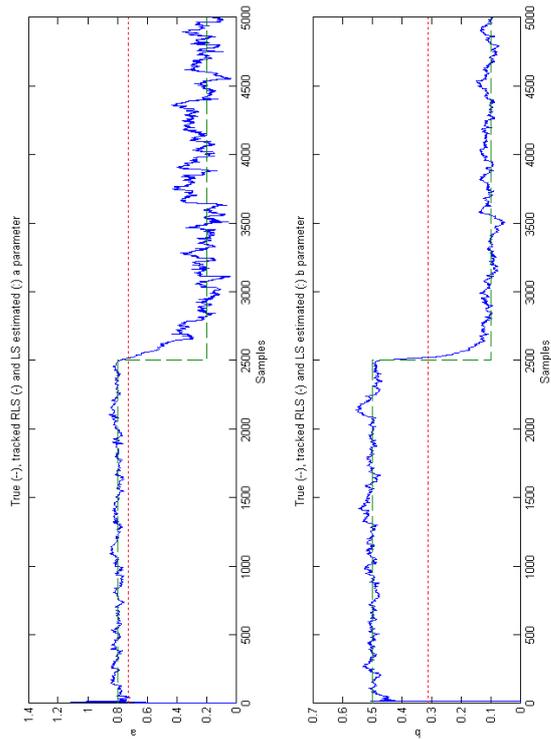
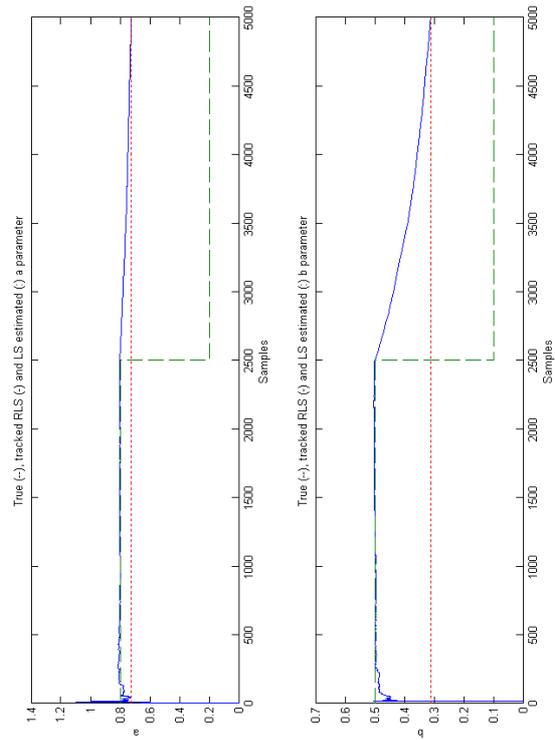
$$\mathbf{P}(t) = \frac{1}{\lambda(t)} \left[ \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)}{\lambda(t) + \boldsymbol{\varphi}^T(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} \right]$$

$\lambda = 0.90$



$\lambda = 0.97$



$\lambda = 0.98$  $\lambda = 1.0$ 

### Initial Conditions

- $\hat{\boldsymbol{\theta}}(0)$  is the initial parameter estimate.
- View  $\boldsymbol{P}(0)$  as an estimate of the covariance matrix of the initial parameter estimate.
  - $\boldsymbol{P}(0)$  (and  $\boldsymbol{P}(t)$ ) are covariance matrices, and must be symmetric and positive definite.
  - Choose  $\boldsymbol{P}(0) = \rho \boldsymbol{I}$ .
  - $\rho$  large  $\Rightarrow$  large initial response. Good if initial estimate  $\hat{\boldsymbol{\theta}}(0)$  is uncertain.

## Forgetting Factor

Let  $\lambda(t) = \lambda$ . The forgetting factor  $\lambda$  will then determine the tracking capability.

- We must have  $\lambda = 1$  to get convergence.
- $\lambda$  small  $\Rightarrow$  Old data is forgotten faster, hence better tracking.
- $\lambda$  small  $\Rightarrow$  the algorithm is more sensitive to noise (bad convergence).
- The memory constant is defined as  $T_0 = \frac{1}{1-\lambda}$ . If  $\lambda = 0.95$ ,  $T_0 = 20$

The choice of  $\lambda$  is consequently a trade-off between tracking capability and noise sensitivity. A typical choice is  $\lambda \in (0.95, 0.99)$ . It is common to let  $\lambda(t)$  tend exponentially to 1, *e.g.*,

$$\lambda(t) = 1 - \lambda_0^t(1 - \lambda(0))$$

## Common Problems for Recursive Identification

- Persistent excitation.
- Estimator windup.
- $P(t)$  becomes indefinite.

## Persistent Excitation

Just as for the off-line case, it is important that the input signal changes sufficiently in order to excite the system so that the experimental data contains enough information about the dynamics of the system. This leads to the concept of *persistent excitation* relating to the input signal. Such persistent excitation concept/condition applies during the whole identification period.

Most stable linear systems may be represented by the so called finite impulse response (FIR) model:

$$y(t) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_n u(t-n) = \varphi^T(t) \theta$$

where

$$\theta = [b_1 \ \dots \ b_n]^T$$

$$\varphi = [u(t-1) \ \dots \ u(t-n)]^T$$

The estimate is given by  $\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y$ , where

$$\Phi^T \Phi = \begin{bmatrix} \sum_{k=n+1}^t u(k-1)^2 & \sum_{k=n+1}^t u(k-1)u(k-2) & \dots & \sum_{k=n+1}^t u(k-1)u(k-n) \\ \sum_{k=n+1}^t u(k-2)u(k-1) & \sum_{k=n+1}^t u(k-2)^2 & \dots & \sum_{k=n+1}^t u(k-2)u(k-n) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=n+1}^t u(k-n)u(k-1) & \sum_{k=n+1}^t u(k-n)u(k-2) & \dots & \sum_{k=n+1}^t u(k-n)^2 \end{bmatrix}$$

This matrix has to be non-singular for the estimate to be unique. This is called an excitation condition. For a long data set,  $t \rightarrow \infty$ , and all sums may be taken from 1 to  $t$ .

Define:

$$C_n = \lim_{t \rightarrow \infty} \frac{1}{t} \Phi^T \Phi = \begin{bmatrix} c(0) & c(1) & \cdots & c(n-1) \\ c(1) & c(0) & \cdots & c(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ c(n-1) & c(n-2) & \cdots & c(0) \end{bmatrix}$$

where  $c(k)$  are the empirical covariances of the input. That is:

$$c(k) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum_{i=1}^t u(i)u(i-k)$$

**Definition:** A signal  $u$  is called *persistently exciting* (PE) of order  $n$  if the matrix  $C_n$  is positive definite.

### Examples:

- **White noise:**  $u(t)$  is white noise, with zero mean and variance  $\sigma^2$ . Then  $c(n) = \sigma^2 \delta_n$ , and  $C_n = \sigma^2 I_n$ , which is always positive definite. Thus  $C_n$  is nonsingular for all  $n$ , and white noise signal is PE of all orders.
- **Step signal:**  $u(t)$  is a step of magnitude  $\sigma$ , then  $c(k) = \sigma^2$ , and  $C_n$  is nonsingular only if  $n = 1$ . Then a step is PE of order 1.
- **Impulse signal:**  $u(t) = 1$  for  $t = 0$ , and 0 otherwise. This gives  $c(n) = 0$  for all  $n$  and  $C_n = 0$ . Therefore, this signal is not PE of any order.

**Important Note:** It is necessary for consistent estimation of an  $n$ -th order system that the input signal be at least persistently exciting of order  $2n$ .

## Conclusions

- In practical scenarios, one often need to use recursive identification (time-varying systems, online identification, fault diagnosis).
- Both the LS and the IVM can easily be recast in recursive forms. The PEM can only be approximated.
- The properties of the on-line methods are comparable with the off-line case.
- Tracking capability can be incorporated by using forgetting factor techniques, or by model the parameter variations.
- There is always a tradeoff between convergence speed and tracking properties, as well as computational complexity and accuracy.
- In practice, one can make simplifications and modifications to make the recursion cheaper and more numerically robust.