

System Identification and Recursive Least Squares

Xu Chen

University of Washington

Motivation

We have been assuming knowledge of the plant in controller design. In practice, plant models come from:

- ▶ modeling by physics: Newton's law, conservation of energy, etc
- ▶ (input-output) data-based system identification

The need for system identification and adaptive control comes from

- ▶ unknown plants
- ▶ time-varying plants
- ▶ known disturbance structure but unknown disturbance parameters

System modeling

Consider the input-output relationship of a plant:

$$u(k) \longrightarrow \boxed{G_p(z^{-1}) = \frac{z^{-1}B(z^{-1})}{A(z^{-1})}} \longrightarrow y(k)$$

or equivalently

$$u(k) \longrightarrow \boxed{\frac{B(z^{-1})}{A(z^{-1})}} \longrightarrow y(k+1) \quad (1)$$

where

$$B(z^{-1}) = b_0 + b_1 z^{-1} + \dots + b_m z^{-m}; \quad A(z^{-1}) = 1 + a_1 z^{-1} + \dots + a_n z^{-n}$$

- $y(k+1)$ is a linear combination of $y(k), \dots, y(k+1-n)$ and $u(k), \dots, u(k-m)$:

$$y(k+1) = - \sum_{i=1}^n a_i y(k+1-i) + \sum_{i=0}^m b_i u(k-i) \quad (2)$$

System modeling

Define *parameter vector* θ and *regressor vector* $\phi(k)$:

$$\theta \triangleq [a_1, a_2, \dots, a_n, b_0, b_1, \dots, b_m]^T$$

$$\phi(k) \triangleq [-y(k), \dots, -y(k+1-n), u(k), u(k-1), \dots, u(k-m)]^T$$

► (2) can be simply written as:

$$\boxed{y(k+1) = \phi^T(k) \theta} \quad (3)$$

- $\phi(k)$ and $y(k+1)$ are known or measured
- **goal**: estimate the unknown θ

Parameter estimation

Suppose we have an estimate of the parameter vector:

$$\hat{\theta} \triangleq [\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n, \hat{b}_0, \hat{b}_1, \dots, \hat{b}_m]^T$$

At time k , we can do estimation:

$$\boxed{\hat{y}(k+1) = \phi^T(k) \hat{\theta}(k)} \quad (4)$$

where $\hat{\theta}(k) \triangleq [\hat{a}_1(k), \hat{a}_2(k), \dots, \hat{a}_n(k), \hat{b}_0(k), \hat{b}_1(k), \dots, \hat{b}_m(k)]^T$

Parameter identification by least squares (LS)

At time k , the least squares (LS) estimate of θ minimizes:

$$J_k = \sum_{i=1}^k \left[y(i) - \phi^T(i-1) \hat{\theta}(k) \right]^2 \quad (5)$$

Solution:

$$J_k = \sum_{i=1}^k \left(y(i)^2 + \hat{\theta}^T(k) \phi(i-1) \phi^T(i-1) \hat{\theta}(k) - 2y(i) \phi^T(i-1) \hat{\theta}(k) \right)$$

Letting $\partial J_k / \partial \hat{\theta}(k) = 0$ yields

$$\hat{\theta}(k) = \underbrace{\left[\sum_{i=1}^k \phi(i-1) \phi^T(i-1) \right]^{-1}}_{F(k)} \sum_{i=1}^k \phi(i-1) y(i) \quad (6)$$

Parameter identification by least squares (LS)

At time $k+1$, we receive a new input and we have one more measurement $y(k+1)$.

Instead of (5), we can do better by minimizing

$$J_{k+1} = \sum_{i=1}^{k+1} \left[y(i) - \phi^T(i-1) \hat{\theta}(k+1) \right]^2$$

whose solution is

$$\hat{\theta}(k+1) = \overbrace{\left[\sum_{i=1}^{k+1} \phi(i-1) \phi^T(i-1) \right]^{-1}}^{F(k+1)} \sum_{i=1}^{k+1} \phi(i-1) y(i) \quad (7)$$

From LS to recursive least squares (RLS)

At time k ,

$$J_k = \sum_{i=1}^k \left[y(i) - \phi^T(i-1) \hat{\theta}(k) \right]^2 \quad (8)$$

$$\hat{\theta}(k) = \left[\sum_{i=1}^k \phi(i-1) \phi^T(i-1) \right]^{-1} \sum_{i=1}^k \phi(i-1) y(i) \quad (9)$$

At time $k+1$,

$$J_{k+1} = \sum_{i=1}^{k+1} \left[y(i) - \phi^T(i-1) \hat{\theta}(k+1) \right]^2$$
$$\hat{\theta}(k+1) = \left[\sum_{i=1}^{k+1} \phi(i-1) \phi^T(i-1) \right]^{-1} \sum_{i=1}^{k+1} \phi(i-1) y(i) \quad (10)$$

recursive least squares (RLS): no need to directly compute (10), $\hat{\theta}(k+1)$ can be obtained by

$$\hat{\theta}(k+1) = \hat{\theta}(k) + [\text{correction term}]$$

(11)

RLS parameter adaptation

Goal: to obtain

$$\hat{\theta}(k+1) = \hat{\theta}(k) + [\text{correction term}] \quad (12)$$

Observation: common building blocks exist:

$$\hat{\theta}(k) = \left(\sum_{i=1}^k \phi(i-1) \phi^T(i-1) \right)^{-1} \boxed{\sum_{i=1}^k \phi(i-1) y(i)}$$

$$\hat{\theta}(k+1) = \left(\sum_{i=1}^k \phi(i-1) \phi^T(i-1) + \phi(k) \phi^T(k) \right)^{-1} \left(\boxed{\sum_{i=1}^k \phi(i-1) y(i)} + \phi(k) y(k+1) \right)$$

RLS parameter adaptation

Goal:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + [\text{correction term}]$$

Derivations:

$$\begin{aligned}\hat{\theta}(k+1) &= F(k+1) \sum_{i=1}^{k+1} \phi(i-1)y(i) \\ &= F(k+1) \left[\sum_{i=1}^k \phi(i-1)y(i) + \phi(k)y(k+1) \right] \\ &= F(k+1) \left[\underline{F(k)^{-1} \hat{\theta}(k)} + \phi(k)y(k+1) \right]\end{aligned}$$

due to

$$\hat{\theta}(k) = F(k) \sum_{i=1}^k \phi(i-1)y(i) \Rightarrow \sum_{i=1}^k \phi(i-1)y(i) = F(k)^{-1} \hat{\theta}(k)$$

RLS parameter adaptation

Goal:

$$\hat{\theta}(k+1) = \hat{\theta}(k) + [\text{correction term}]$$

Derivations (Cont'd): noting now

$$F(k+1)^{-1} = \sum_{i=1}^{k+1} \phi(i-1)\phi^T(i-1) = F(k)^{-1} + \phi(k)\phi^T(k)$$

we have

$$\begin{aligned}\hat{\theta}(k+1) &= F(k+1) \left[F(k)^{-1} \hat{\theta}(k) + \phi(k)y(k+1) \right] \\ &= F(k+1) \left[\left(F(k+1)^{-1} - \phi(k)\phi^T(k) \right) \hat{\theta}(k) + \phi(k)y(k+1) \right] \\ &= \hat{\theta}(k) + F(k+1)\phi(k) \left[y(k+1) - \phi^T(k)\hat{\theta}(k) \right]\end{aligned}\tag{13}$$

RLS parameter adaptation

Define

$$\begin{aligned}\hat{y}^o(k+1) &= \phi^T(k)\hat{\theta}(k) \\ \varepsilon^o(k+1) &= y(k+1) - \hat{y}^o(k+1)\end{aligned}$$

(13) is equivalent to

$$\boxed{\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)\varepsilon^o(k+1)} \quad (14)$$

RLS adaptation gain recursion

$F(k+1)$, called the adaptation gain, can be updated recursively too!:

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)} \quad (15)$$

Proof:

- ▶ matrix inversion lemma (MIL): if A is nonsingular, B and C have compatible dimensions, then

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(CA^{-1}B + I)^{-1}CA^{-1}$$

- ▶ note the algebra

$$\begin{aligned} F(k+1) &= \left[\sum_{i=1}^{k+1} \phi(i-1)\phi^T(i-1) \right]^{-1} = \left[F(k)^{-1} + \phi(k)\phi^T(k) \right]^{-1} \\ &\stackrel{\text{MIL}}{=} F(k) - F(k)\phi(k) \left(\phi^T(k)F(k)\phi(k) + 1 \right)^{-1} \phi^T(k)F(k) \end{aligned}$$

which gives (15)

RLS parameter adaptation

The recursive adaptation gain yields an adaptation law using only the past data:

$$\begin{aligned}(15) \Rightarrow F(k+1)\phi(k) &= F(k)\phi(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)}\phi(k) \\ &= \frac{F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)}\end{aligned}$$

Hence we have the parameter adaptation algorithm (PAA):

$$\begin{aligned}\hat{\theta}(k+1) &= \hat{\theta}(k) + F(k+1)\phi(k)\varepsilon^o(k+1) \\ &= \hat{\theta}(k) + \frac{F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)}\varepsilon^o(k+1) \\ F(k+1) &= F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)}\end{aligned}$$

PAA implementation: initial values

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \frac{F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)} \varepsilon^o(k+1)$$

- ▶ $\hat{\theta}(0)$: initial guess of parameter vector
- ▶ $F(k)$ is symmetric by definition:

$$F(k) = \left[\sum_{i=1}^k \phi(i-1)\phi^T(i-1) \right]^{-1}$$

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)}$$

- ▶ $F(0)$: commonly set as $F(0) = \sigma I_{n+m+1}$, where σ is a large number and I_{n+m+1} is an $n+m+1$ -row identity matrix, as $F(k)$ is always non-increasing:

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1 + \phi^T(k)F(k)\phi(k)}$$

RLS parameter adaptation

Up till now we have been using the *a priori* prediction and *a priori* prediction error:

$$\begin{aligned}\hat{y}^o(k+1) &= \phi^T(k)\hat{\theta}(k) : \text{after measurement of } y(k) \\ \varepsilon^o(k+1) &= y(k+1) - \hat{y}^o(k+1)\end{aligned}$$

Further analysis (e.g., convergence of $\hat{\theta}(k)$) requires the new definitions of *a posteriori* prediction and *a posteriori* prediction error:

$$\begin{aligned}\hat{y}(k+1) &= \phi^T(k)\hat{\theta}(k+1) : \text{after measurement of } y(k+1) \\ \varepsilon(k+1) &= y(k+1) - \hat{y}(k+1)\end{aligned}$$

Relationship between $\varepsilon(k+1)$ and $\varepsilon^o(k+1)$

Note that

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \frac{F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)}\varepsilon^o(k+1)$$

$$\Rightarrow \underbrace{\phi^T(k)\hat{\theta}(k+1)}_{\hat{y}(k+1)} = \underbrace{\phi^T(k)\hat{\theta}(k)}_{\hat{y}^o(k+1)} + \frac{\phi^T(k)F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)}\varepsilon^o(k+1)$$

$$\Rightarrow \underbrace{y(k+1) - \hat{y}(k+1)}_{\varepsilon(k+1)} = \underbrace{y(k+1) - \hat{y}^o(k+1)}_{\varepsilon^o(k+1)} - \frac{\phi^T(k)F(k)\phi(k)}{1 + \phi^T(k)F(k)\phi(k)}\varepsilon^o(k+1)$$

Hence

$$\boxed{\varepsilon(k+1) = \frac{\varepsilon^o(k+1)}{1 + \phi^T(k)F(k)\phi(k)}} \quad (16)$$

- note: $|\varepsilon(k+1)| \leq |\varepsilon^o(k+1)|$ ($\hat{y}(k+1)$ is more accurate than $\hat{y}^o(k+1)$)

A posteriori RLS parameter adaptation

With (16), we can write the PAA in the *a posteriori* form

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k)\phi(k)\varepsilon(k+1) \quad (17)$$

which is not implementable but is needed for stability analysis.

Forgetting factor

motivation

- ▶ previous discussions assume the actual parameter vector θ is constant
- ▶ adaptation gain $F(k)$ keeps decreasing, as $F^{-1}(k)$ keeps increasing:

$$F^{-1}(k+1) = F^{-1}(k) + \phi(k)\phi^T(k)$$

- ▶ hence the adaptation becomes weaker and weaker as k increases
- ▶ for time-varying parameters, we need a mechanism to “forget” the “old” data

Forgetting factor

Consider a new cost

$$J_k = \sum_{i=1}^k \lambda^{k-i} \left[y(i) - \phi^T(i-1) \hat{\theta}(k) \right]^2, \quad 0 < \lambda \leq 1$$

► past errors are less weighted:

$$\begin{aligned} J_k = & \left[y(k) - \phi^T(k-1) \hat{\theta}(k) \right]^2 + \lambda \left[y(k-1) - \phi^T(k-2) \hat{\theta}(k) \right]^2 \\ & + \lambda^2 \left[y(k-2) - \phi^T(k-3) \hat{\theta}(k) \right]^2 + \dots \end{aligned}$$

► the solution is:

$$\hat{\theta}(k) = \overbrace{\left[\sum_{i=1}^k \lambda^{k-i} \phi(i-1) \phi^T(i-1) \right]^{-1}}^{F(k)} \sum_{i=1}^k \lambda^{k-i} \phi(i-1) y(i) \quad (18)$$

Forgetting factor

- ▶ in (18), the recursion of the adaptation gain is:

$$F(k+1)^{-1} = \lambda F(k)^{-1} + \phi(k)\phi(k)^T$$

- ▶ or, equivalently, after using matrix inversion lemma:

$$F(k+1) = \frac{1}{\lambda} \left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\lambda + \phi^T(k)F(k)\phi(k)} \right] \quad (19)$$

Forgetting factor

- ▶ in (18), the recursion of the adaptation gain is:

$$F(k+1)^{-1} = \lambda F(k)^{-1} + \phi(k)\phi(k)^T$$

- ▶ or, equivalently, after using matrix inversion lemma:

$$F(k+1) = \frac{1}{\lambda} \left[F(k) - \frac{F(k)\phi(k)\phi(k)^T F(k)}{\lambda + \phi(k)^T F(k)\phi(k)} \right] \quad (19)$$

Forgetting factor

The weighting can be made more flexible:

$$F(k+1) = \frac{1}{\lambda_1(k)} \left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\lambda_1(k) + \phi^T(k)F(k)\phi(k)} \right]$$

which corresponds to the cost function

$$J_k = \left[y(k) - \phi^T(k-1)\hat{\theta}(k) \right]^2 + \lambda_1(k-1) \left[y(k-1) - \phi^T(k-2)\hat{\theta}(k) \right]^2 \\ + \lambda_1(k-1)\lambda_1(k-2) \left[y(k-2) - \phi^T(k-3)\hat{\theta}(k) \right]^2 + \dots$$

i.e. (note that $\prod_{j=k}^{k-1} \lambda_1(j) \triangleq 1$)

$$J_k = \sum_{i=1}^k \left\{ \left(\prod_{j=i}^{k-1} \lambda_1(j) \right) \left[y(i) - \phi^T(i-1)\hat{\theta}(k) \right]^2 \right\}$$

Forgetting factor

The general form of the adaptation gain is:

$$F(k+1) = \frac{1}{\lambda_1(k)} \left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)} + \phi^T(k)F(k)\phi(k)} \right] \quad (20)$$

which comes from:

$$F(k+1)^{-1} = \lambda_1(k)F(k)^{-1} + \lambda_2(k)\phi(k)\phi^T(k)$$

with $0 < \lambda_1(k) \leq 1$ and $0 \leq \lambda_2(k) \leq 2$ (for stability requirements, will come back to this soon).

$\lambda_1(k)$	$\lambda_2(k)$	PAA
1	0	constant adaptation gain
1	1	least square gain
< 1	1	least square gain with forgetting factor

*Influence of the initial conditions

If we initialize $F(k)$ and $\hat{\theta}(k)$ at F_0 and θ_0 , we are actually minimizing

$$J_k = \left(\hat{\theta}(k) - \theta_0 \right)^T F_0^{-1} \left(\hat{\theta}(k) - \theta_0 \right) + \sum_{i=1}^k \alpha_i \left[y(i) - \phi^T(i-1) \hat{\theta}(k) \right]^2$$

where α_i is the weighting for the error at time i . The least square parameter estimate is

$$\hat{\theta}(k) = \left[F_0^{-1} + \sum_{i=1}^k \alpha_i \phi(i-1) \phi^T(i-1) \right]^{-1} \left[F_0^{-1} \theta_0 + \sum_{i=1}^k \alpha_i \phi(i-1) y(i) \right]$$

We see the relative importance of the initial values decays with time.

*Influence of the initial conditions

If it is possible to wait a few samples before the adaptation, proper initial values can be obtained if the recursion is started at time k_0 with

$$F(k_0) = \left[\sum_{i=1}^{k_0} \alpha_i \phi(i-1) \phi^T(i-1) \right]^{-1}$$
$$\hat{\theta}(k_0) = F(k_0) \sum_{i=1}^{k_0} \alpha_i \phi(i-1) y(i)$$

*Matrix version of the least squares solution

the least squares solution targets to minimize

$$J_k = \sum_{i=1}^k e^o(i)^2 = \sum_{i=1}^k \left[y(i) - \phi^T(i-1) \hat{\theta}(k) \right]^2$$

putting the above error term in a vector form yields

$$\begin{bmatrix} e^o(k) \\ \vdots \\ e^o(1) \end{bmatrix} = \underbrace{\begin{bmatrix} y(k) \\ \vdots \\ y(1) \end{bmatrix}}_{Y_k} - \underbrace{\begin{bmatrix} \phi(k-1) & \phi(k-2) & \dots & \phi(0) \end{bmatrix}^T}_{\Phi_{k-1}^T: \text{a skinny matrix, } k \times (n_A + n_B)} \hat{\theta}(k)$$

hence

$$\begin{aligned} J_k &= \|Y_k - \Phi_{k-1}^T \hat{\theta}(k)\|^2 \\ &= \left(\hat{\theta}^T(k) \Phi_{k-1}^T \Phi_{k-1} \hat{\theta}(k) - 2 Y_k^T \Phi_{k-1}^T \hat{\theta}(k) + Y_k^T Y_k \right) \end{aligned}$$

*Matrix version of the least squares solution

with $J_k = \left(\hat{\theta}^T(k) \Phi_{k-1}^T \Phi_{k-1} \hat{\theta}(k) - 2Y_k^T \Phi_{k-1}^T \hat{\theta}(k) + Y_k^T Y_k \right)$,
taking the partial derivative yields

$$\frac{\partial J_k}{\partial \theta} = 2\Phi_{k-1}^T \Phi_{k-1} \hat{\theta}(k) - 2\Phi_{k-1}^T Y_k$$

by letting $\frac{\partial J_k}{\partial \theta} = 0$, we obtain the unique optimal solution:

$$\hat{\theta}(k)^* = \left(\Phi_{k-1}^T \Phi_{k-1} \right)^{-1} \Phi_{k-1}^T Y_k$$

the optimal error vector is

$$e_k = Y_k - \Phi_{k-1}^T \hat{\theta}(k)^* = \left(I - \Phi_{k-1}^T \left(\Phi_{k-1}^T \Phi_{k-1} \right)^{-1} \Phi_{k-1} \right) Y_k$$

and the optimal cost is

$$J_k^* = \|Y_k - \Phi_{k-1}^T \left(\Phi_{k-1}^T \Phi_{k-1} \right)^{-1} \Phi_{k-1}^T Y_k\|^2$$