System Identification and Recursive Least Squares

Big picture

We have been assuming knwoledge 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 come from

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

System modeling

Consider the input-output relationship of a plant:

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

or equivalently

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

where

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

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

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

System modeling

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

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

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

- (2) can be simply written as:

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

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

Parameter estimation

Suppose we have an estimate of the parameter vector:

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

At time k, we can do estimation:

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

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

Parameter identification by least squares (LS)

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

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

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

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

Recursive least squares (RLS)

At time $k+1$, we know $u(k+1)$ and have one more measurement $y(k+1)$.
Instead of (5), we can do better by minimizing

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

whose solution is

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

recursive least squares (RLS): no need to do the matrix inversion in (7), $\hat{\theta}(k+1)$ can be obtained by

$$\boxed{\hat{\theta}(k+1) = \hat{\theta}(k) + [\text{correction term}]} \tag{8}$$

RLS parameter adaptation

Goal: to obtain $\boxed{\hat{\theta}(k+1) = \hat{\theta}(k) + [\text{correction term}]}$ (9)

Derivations:

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

$$\begin{aligned}
\hat{\theta}(k+1) &= F(k+1) \sum_{i=1}^{k+1} \phi(i-1) y(i) \\
&= F(k+1) \left[\sum_{i=1}^{k} \phi(i-1) y(i) + \phi(k) y(k+1) \right] \\
&= F(k+1) \left[F(k)^{-1} \hat{\theta}(k) + \phi(k) y(k+1) \right] \\
&= F(k+1) \left[\left(F(k+1)^{-1} - \phi(k)\phi^T(k) \right) \hat{\theta}(k) + \phi(k) y(k+1) \right] \\
&= \hat{\theta}(k) + F(k+1)\phi(k) \left[y(k+1) - \hat{\theta}^T(k)\phi(k) \right] \qquad (10)
\end{aligned}$$

RLS parameter adaptation

Define

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

(10) is equivalent to

$$\boxed{\hat{\theta}(k+1)=\hat{\theta}(k)+F(k+1)\phi(k)\varepsilon^o(k+1)} \qquad (11)$$

RLS adaptation gain recursion

$F(k+1)$ is called the adaptation gain, and can be updated by

$$\boxed{F(k+1)=F(k)-\frac{F(k)\phi(k)\phi^T(k)F(k)}{1+\phi^T(k)F(k)\phi(k)}} \qquad (12)$$

Proof:

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

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

- note the algebra

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

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

which gives (12)

RLS parameter adaptation

An alternative representation of adaptation law (11):

$$(12) \Rightarrow F(k+1)\phi(k) = F(k)\phi(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1+\phi^T(k)F(k)\phi(k)}\phi(k)$$
$$= \frac{F(k)\phi(k)}{1+\phi^T(k)F(k)\phi(k)}$$

Hence we have the parameter adaptation algorithm (PAA):

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

PAA implementation

- $\hat{\theta}(0)$: initial guess of parameter vector

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

- $F(0) = \sigma I$: σ is a large number, as $F(k)$ is always none-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:

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

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

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

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

$$\boxed{\varepsilon(k+1)=\frac{\varepsilon^o(k+1)}{1+\phi^T(k)F(k)\phi(k)}} \qquad (13)$$

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

A posteriori RLS parameter adaptation

With (13), we can write the PAA in the *a posteriori* form

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

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

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

- this means adaptation becomes weaker and weaker
- for time-varying parameters, we need a mechanism to "forget" the "old' data

Forgetting factor

Consider a new cost

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

- past errors are less weighted:

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

- the solution is:

$$\hat{\theta}(k) = \overbrace{\left[\sum_{i=1}^{k} \lambda^{k-i}\phi(i-1)\phi^T(i-1) \right]^{-1}}^{F(k)} \sum_{i=1}^{k} \lambda^{k-i}\phi(i-1)y(i) \quad (15)$$

Forgetting factor

- in (15), the recursion of the adaptation gain is:

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

or, equivalently

$$F(k+1) = \frac{1}{\lambda} \left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\lambda + \phi^T(k)F(k)\phi(k)} \right] \quad (16)$$

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

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

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

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

Forgetting factor

The general form of the adaptation gain is:

$$F(k+1) = \frac{1}{\lambda_1(k)}\left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{\frac{\lambda_1(k)}{\lambda_2(k)}+\phi^T(k)F(k)\phi(k)}\right] \quad (17)$$

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

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

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

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

$$\hat{\theta}(k_0) = F(k_0)\sum_{i=1}^{k_0} \alpha_i \phi(i-1)y(i)$$