EECE 574 - Adaptive Control

Recursive Identification Algorithms

Guy Dumont

Department of Electrical and Computer Engineering
University of British Columbia

January 2010

Recursive Identification

- There are many situations when it is preferable to perform the identification on-line, such as in **adaptive control**.
- Identification methods need to be implemented in a recursive fashion, i.e. the parameter estimate at time t should be computed as a function of the estimate at time $t-1$ and of the incoming information at time t.
- Recursive least-squares.
- Recursive instrumental variables.
- Recursive extended least-squares and recursive maximum likelihood.

Recursive Least-Squares (RLS)

We have seen that, with t observations available, the least-squares estimate is

$$\hat{\theta}(t) = [X^T(t)X(t)]^{-1}X^T(t)Y(t)$$

with

$$Y^T(t) = [\ y(1) \quad \cdots \quad y(t)]$$

$$X(t) = \begin{bmatrix} x^T(1) \\ \vdots \\ x^T(t) \end{bmatrix}$$

Assume one additional observation becomes available, the problem is then to find $\hat{\theta}(t+1)$ as a function of $\hat{\theta}(t)$ and $y(t+1)$ and $u(t+1)$.

Recursive Least-Squares (RLS)

Defining $X(t+1)$ and $\underline{Y}(t+1)$ as

$$X(t+1) = \begin{bmatrix} X(t) \\ x^T(t+1) \end{bmatrix} \qquad Y(t+1) = \begin{bmatrix} Y(t) \\ y(t+1) \end{bmatrix}$$

and defining $P(t)$ and $P(t+1)$ as

$$P(t) = [X^T(t)X(t)]^{-1} \qquad P(t+1) = [X^T(t+1)X(t+1)]^{-1}$$

one can write

$$P(t+1) = [X^T(t)X(t) + x(t+1)x^T(t+1)]^{-1}$$

$$\hat{\theta}(t+1) = P(t+1)[X^T(t)Y(t) + x(t+1)y(t+1)]$$

Matrix Inversion Lemma

Let A, D and $[D^{-1} + CA^{-1}B]$ be nonsingular square matrices. Then $A + BDC$ is invertible and

$$\boxed{(A+BDC)^{-1} = A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1}}$$

Proof The simplest way to prove it is by direct multiplication

$$\begin{aligned}
& (A+BDC)(A^{-1} - A^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1}) \\
= \; & I + BDCA^{-1} - B(D^{-1} + CA^{-1}B)^{-1}CA^{-1} \\
& -BDCA^{-1}B(D^{-1} + CA^{-1}B)^{-1}CA^{-1} \\
= \; & I + BDCA^{-1} - BD(D^{-1} + CA^{-1}B)(D^{-1} + CA^{-1}B)^{-1}CA^{-1} \\
= \; & I
\end{aligned}$$

Matrix Inversion Lemma

An alternative form, useful for deriving recursive least-squares is obtained when B and C are $n \times 1$ and $1 \times n$ (i.e. column and row vectors):

$$(A+BC)^{-1} = A^{-1} - \frac{A^{-1}BCA^{-1}}{1+CA^{-1}B}$$

Now, consider

$$P(t+1) = [X^T(t)X(t) + x(t+1)x^T(t+1)]^{-1}$$

and use the matrix-inversion lemma with

$$A = X^T(t)X(t) \qquad B = x(t+1) \qquad C = x^T(t+1)$$

Recursive Least-Squares (RLS)

Some simple matrix manipulations then give the recursive least-squares algorithm:

RLS

$$
\begin{aligned}
\hat{\theta}(t+1) &= \hat{\theta}(t) + K(t+1)[y(t+1) - x^T(t+1)\hat{\theta}(t)] \\
K(t+1) &= \frac{P(t)x(t+1)}{1 + x^T(t+1)P(t)x(t+1)} \\
P(t+1) &= P(t) - \frac{P(t)x(t+1)x^T(t+1)P(t)}{1 + x^T(t+1)P(t)x(t+1)}
\end{aligned}
$$

Note that $K(t+1)$ can also be expressed as

$$K(t+1) = P(t+1)x(t+1)$$

Recursive Least-Squares (RLS)

- The recursive least-squares algorithm is the **exact** mathematical equivalent of the batch least-squares.
- Once initialized, no matrix inversion is needed.
- Matrices stay the same size all the time.
- Computationally very efficient.
- P is proportional to the covariance matrix of the estimate, and is thus called the covariance matrix.
- The algorithm has to be initialized with $\hat{\theta}(0)$ and $P(0)$. Generally, $P(0)$ is initialized as αI where I is the identity matrix and α is a large positive number. The larger α, the less confidence is put in the initial estimate $\hat{\theta}(0)$.

RLS and Kalman Filter

There are some very strong connections between the recursive least-squares algorithm and the Kalman filter. Indeed, the RLS algorithm has the structure of a Kalman filter:

$$\underbrace{\hat{\theta}(t+1)}_{\text{new}} = \underbrace{\hat{\theta}(t)}_{\text{old}} + K(t+1)\underbrace{\left[y(t+1) - x^T(t+1)\hat{\theta}(t)\right]}_{\text{correction}}$$

where $K(t+1)$ is the Kalman gain.

Matlab Implementation

The following Matlab code is a straightforward implementation of the RLS algorithm:

```
function [thetaest,P]=rls(y,x,thetaest,P)
% RLS
% y,x: current measurement and regressor
% thetaest, P: parameter estimates and covariance matrix
K= P*x/(1+x'*P*x); % Gain
P= P- (P*x*x'*P)/(1+x'*P*x); % Covariance matrix update
thetaest= thetaest +K*(y-x'*thetaest); %Estimate update
end
```

Recursive Extended Least-Squares and Recursive Maximum-Likelihood

Because the prediction error is not linear in the C-parameters, it is not possible to derive an exact recursive maximum likelihood method as for the least-squares method.

The ARMAX model

$$A(q^{-1})y(t) = B(q^{-1})u(t) + C(q^{-1})e(t)$$

can be written as

$$y(t) = \underline{x}^T(t)\underline{\theta} + e(t)$$

with

$$\begin{aligned} \theta &= [a_1, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_n]^T \\ x^T(t) &= [-y(t-1), \ldots, -y(t-n), u(t-1), \\ &\quad \ldots, u(t-n), e(t-1), \ldots, e(t-n)]^T \end{aligned}$$

Recursive Extended Least-Squares and Approximate Maximum-Likelihood

- If $e(t)$ was known, RLS could be used to estimate θ, however it is unknown and thus has to be estimated.
- It can be done in two ways, either using the prediction error or the residual.
- The first case corresponds to the RELS method, the second to the AML method.

Recursive Extended Least-Squares and Approximate Maximum-Likelihood

- The one-step ahead prediction error is defined as

$$\begin{aligned} \varepsilon(t) &= y(t) - \hat{y}(t \mid t-1) \\ &= y(t) - x^T(t)\hat{\theta}(t-1) \end{aligned}$$

$$x(t) = [-y(t-1), \ldots, u(t-1), \ldots, \varepsilon(t-1), \ldots, \varepsilon(t-n)]^T$$

- The residual is defined as

$$\begin{aligned} \eta(t) &= y(t) - \hat{y}(t \mid t) \\ &= y(t) - x^T(t)\hat{\theta}(t) \end{aligned}$$

$$x(t) = [-y(t-1), \ldots, u(t-1), \ldots, \eta(t-1), \ldots, \eta(t-n)]^T$$

Recursive Extended Least-Squares and Approximate Maximum-Likelihood

- Sometimes $\varepsilon(t)$ and $\eta(t)$ are also referred to as a-priori and a-posteriori prediction errors.
- Because it uses the latest estimate $\hat{\theta}(t)$, as opposed to $\hat{\theta}(t-1)$ for $\varepsilon(t)$, $\eta(t)$ is a better estimate, especially in transient behaviour.
- Note however that if $\hat{\theta}(t)$ converges as $t \longrightarrow \infty$ then $\eta(t) \longrightarrow \varepsilon(t)$.

Recursive Extended Least-Squares and Approximate Maximum-Likelihood

The two schemes are then described by

$$\begin{aligned}
\hat{\theta}(t+1) &= \hat{\theta}(t) + K(t+1)[y(t+1) - x^T(t+1)\hat{\theta}(t)] \\
K(t+1) &= P(t+1)x(t+1)/[1 + x^T(t+1)P(t)x(t+1)] \\
P(t+1) &= P(t) - \frac{P(t)x(t+1)x^T(t+1)P(t)}{[1 + x^T(t+1)P(t)x(t+1)]}
\end{aligned}$$

but differ by their definition of $x(t)$

Recursive Extended Least-Squares and Approximate Maximum-Likelihood

- The RELS algorithm corresponds uses the prediction error. This algorithm is called RELS, Extended Matrix or RML1 in the literature. It has generally good convergence properties, and has been proved consistent for moving-average and first-order auto regressive processes. However, counterexamples to general convergence exist, see for example Ljung (1975).
- The AML algorithm uses the residual error. The AML has better convergence properties than the RML, and indeed convergence can be proven under rather unrestrictive conditions.

Recursive Maximum-Likelihood

- The ML can also be interpreted in terms of data filtering.
- Consider the performance index:

$$V(t) = \frac{1}{2}\sum_{i=1}^{t} \varepsilon^2(i)$$

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

- Define the filtered regressor $x_f(t)$ as

$$x_f(t) = \frac{1}{\hat{C}(q^{-1})}x(t)$$

- Requires initialization and a stable $\hat{C}(q^{-1})$.

Recursive Maximum-Likelihood

The resulting scheme is then:

$$\begin{aligned}
\hat{\theta}(t+1) &= \hat{\theta}(t) + K(t+1)[y(t+1) - x^T(t+1)\hat{\theta}(t)] \\
K(t+1) &= \frac{P(t+1)x_f(t+1)}{[1 + x_f^T(t+1)P(t)x_f(t+1)]} \\
P(t+1) &= P(t) - \frac{P(t)x_f(t+1)x_f^T(t+1)P(t)}{[1 + x_f^T(t+1)P(t)x_f(t+1)]}
\end{aligned}$$

- No global convergence result available

Properties of AML

Definition

A discrete transfer function is said to be strictly positive real if it is stable and

$$ReH(e^{j\omega}) > 0 \qquad \forall \omega - \pi < \omega \leq \pi$$

on the unit circle.

This condition can be checked by replacing z by $\frac{1+j\omega}{1-j\omega}$ and extracting the real part of the resulting expression.

For the convergence of AML, the following theorem is then available.

Properties of AML

Theorem (Ljung & Söderström, 1983))

Assume both process and model are described by ARMAX with order model $\geq$ order process, then if

1. $\{u(t)\}$ *is sufficiently rich*
2. $\frac{1}{C(q^{-1})} - \frac{1}{2}$ *is strictly positive real*

then $\hat{\theta}(t)$ *will converge such that*

$$E[\varepsilon(t,\underline{\hat{\theta}}) - e(t)]^2 = 0$$

If model and process have the same order, this implies

$$\hat{\theta}(t) \longrightarrow \theta \text{ as } t \longrightarrow \infty$$

A Unified Algorithm

Looking at all the previous algorithms, it is obvious that they all have the same form, with only different parameters. They can all be represented by a recursive prediction - error method (RPEM).

RPEM

$$
\begin{aligned}
\hat{\theta}(t+1) &= \hat{\theta}(t) + K(t+1)\varepsilon(t+1) \\
K(t+1) &= P(t)z(t+1)/[1+x^T(t+1)P(t)z(t+1)] \\
P(t+1) &= P(t) - \frac{P(t)z(t+1)x^T(t+1)P(t)}{[1+x^T(t+1)P(t)x(t+1)]}
\end{aligned}
$$