System Identification, Lecture 7

Kristiaan Pelckmans (IT/UU, 2338)

Course code: 1RT880, Report code: 61800 - Spring 2015
F, FRI Uppsala University, Information Technology

28 April 2015

Lecture 7

- Recursive Identification Methods.
- Recursive Least Squares.
- Tracking case.
- Variations.
- Common themes.

Why?

Why is recursive identification of interest?

- Online estimation.
- Adaptive systems.
- Time-varying Parameters.
- Fault-Detection.

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 *batch* (offline) method, e.g. OLS, PEM.

Desirable Properties

- Fast convergence.
- Consistent estimates (time-invariant case).
- Good tracking (time-varying case).
- Computationally simple (perform all calculations during one sampling interval).

Trade-offs

No free lunch. The design is always based on trade-offs, such as

- Convergence vs. Tracking.
- Computational complexity vs. accuracy.

Recursive Least Squares Method (RLS)

$$\hat{\theta}_t = \operatorname*{argmin}_{\theta} V_t(\theta) \quad V_t(\theta) = \sum_{k=1}^{t} \epsilon_k^2$$

where $\epsilon_k = y_k - \varphi_k^T \theta$. The solution reads as:

$$\hat{\theta}_t = \mathbf{R}_t^{-1} \mathbf{r}_t$$

where

$$\mathbf{R}_t = \sum_{k=1}^{t} \varphi_k \varphi_k^T, \quad \mathbf{r}_t = \sum_{k=1}^{t} \varphi_k y_k$$

- The criterion function $V_t(\theta)$ changes every step, so does $\hat{\theta}_t$
- A 'simple' recursive implementation of $\hat{\theta}_t$?

RLS, Ct'd

Derivation:

$$\hat{\theta}_t = \mathbf{R}_t^{-1}\mathbf{r}_t = \mathbf{R}_t^{-1}\left(\sum_{s=1}^{t-1}\varphi_s y_s + \varphi_t y_t\right)$$
$$= \mathbf{R}_t^{-1}\mathbf{R}_{t-1}\hat{\theta}_{t-1} + \mathbf{R}_t^{-1}(\varphi_t y_t).$$

And since $\mathbf{R}_{t-1} = \mathbf{R}_t - \varphi_t\varphi_t^T$, one has

$$= \mathbf{R}_t^{-1}\left(\mathbf{R}_t - \varphi_t\varphi_t^T\right)\hat{\theta}_{t-1} + \mathbf{R}_t^{-1}(\varphi_t y_t)$$
$$= \hat{\theta}_{t-1} - \mathbf{R}^{-1}\varphi_t\varphi_t^T\hat{\theta}_{t-1} + \mathbf{R}_t^{-1}(\varphi_t y_t)$$
$$= \hat{\theta}_{t-1} + \mathbf{R}_t^{-1}\varphi_t\left(y_t - \varphi_t^T\hat{\theta}_{t-1}\right)$$

RLS, Ct'd

Matrix Inversion Lemma: (assume symmetric, invertible $Z \in \mathbb{R}^{n\times n}$, $z \in \mathbb{R}^n$)

$$Z_+ = Z + zz^T$$

Question: can we write Z_+^{-1} in terms of Z^{-1}?

$$Z_+^{-1} = Z^{-1} - \frac{Z^{-1}zz^TZ^{-1}}{1 + z^TZ^{-1}z}$$

Proof:

$$
\begin{aligned}
&(Z + zz^T)\left(Z^{-1} - \frac{Z^{-1}zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right) \\
&= I_n - Z\left(\frac{Z^{-1}zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right) + zz^TZ^{-1} - (zz^T)\left(\frac{Z^{-1}zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right) \\
&= I_n - \left(\frac{zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right) \\
&+ \left(\frac{zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right)(1 + zZ^{-1}z) - \left(\frac{zz^TZ^{-1}zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right) \\
&= I_n + \left(\frac{zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right)(zz^TZ^{-1}) - \left(\frac{zz^TZ^{-1}zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right)
\end{aligned}
$$

Now, note that $(z^TZ^{-1}z)$ is a scalar, and thus

$$
= I_n + \left(\frac{zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right)(z^TZ^{-1}z) - \left(\frac{(z^TZ^{-1}z)zz^TZ^{-1}}{1 + z^TZ^{-1}z}\right) = I_n.
$$

Q.E.D.

RLS, Ct'd

Recursive algorithm

- At time $t = 0$, choose initial values of $\hat{\theta}(0)$ and $\mathbf{P}(0)$
- For any $t > 0$, compute φ_t and do

$$\begin{cases} \hat{\theta}_t = \hat{\theta}_{t-1} + \mathbf{K}_t \epsilon_t \\ \epsilon_t = y_t - \varphi_t \hat{\theta}_{t-1} \\ \mathbf{K}_t = \mathbf{P}_t \varphi_t \\ \mathbf{P}_t = \left[\mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1} \varphi_t \varphi_t^T \mathbf{P}_{t-1}}{1 + \varphi_t^T \mathbf{P}_{t-1} \varphi_t} \right] \end{cases}$$

Tracking

How to handle time-varying parameter $\theta_{0,t}$?

- Postulate a time-varying model for the parameters. Typically, 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.

$$\hat{\theta}_t = \operatorname*{argmin}_{\theta} V_t(\theta), \quad \text{s.t.} \quad V_t(\theta) = \sum_{k=1}^{t} \beta(t,k)\epsilon_t^2.$$

where the weighting function β satisfies:

$$\begin{cases} \beta(t,k) = \lambda_t \beta(t-1,k), \ 0 < k \leq t \\ \beta(t,t) = 1 \end{cases}$$

A common choice is to let $\lambda_t = \lambda$ with given 'forgetting factor $0 < \lambda \leq 1$, such that

$$\beta(t,k) = \lambda^{t-k}$$

In case $\lambda = 1$, OLS is implemented.

Weighted RLS

Algorithm

- At $t = 0$, choose initial values of $\hat{\theta}_0$ and $\mathbf{P}_0$.
- For each $t > 0$, do

$$\begin{cases} \hat{\theta}_t = \hat{\theta}_{t-1} + \mathbf{K}_t \epsilon_t \\ \epsilon_t = y_t - \varphi_t \hat{\theta}_{t-1} \\ \mathbf{K}_t = \mathbf{P}_t \varphi_t \\ \mathbf{P}_t = \frac{1}{\lambda_t} \left[\mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1}\varphi_t\varphi_t^T\mathbf{P}_{t-1}}{\lambda_t + \varphi_t^T \mathbf{P}_{t-1}\varphi_t} \right] \end{cases}$$

Initial Conditions

- $\hat{\theta}_0$ is the initial estimate (prior).
- View $\mathbf{P}_0$ as the covariance matrix of the initial parameter vector:
  - $\mathbf{P}_0$ are covariance matrices, must be positive definite.
  - Choose $\mathbf{P}_0 = \rho I_n$
  - If ρ large, then large initial response. This is good when initial estimate uncertain.

Forgetting Factor

The forgetting factor λ will set the 'tracking capacity'.

- Consistent if $\lambda = 1$.
- λ small: old data forgotten fastly, good tracking.
- λ small: the algorithm is sensitive to noise - bad convergence.
- The memory constant is $\tau_0 = \frac{1}{1-\lambda}$.

The choice of λ is consequently a trade-off between tracking capability, and noise sensitivity. A typical choice is $\lambda \in [0.95, 0.99[$. Also, it is common to let λ_t tend exponentially to one (why?)

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

The Kalman filter

Consider the (MISO) system

$$\begin{cases} x_{t+1} = \mathbf{F} x_t + \mathbf{G} u_t + v_t \\ y_t = \mathbf{H} x_t + e_t \end{cases}$$

where v_t and e_t are independent white noise sources with $E[e_s^2] = R_2$ and $E[v_t v_t^T] = R_1$.

The optimal predictor of the state variable x_{t+1} based on x_t and the output observation y_t is given by the Kalman filter.

$$\begin{cases} \hat{x}_{t+1} = \mathbf{F} x_t + \mathbf{G} u_t + \mathbf{K}_{t+1} \left(y_{t+1} - \mathbf{H} x_t \right) \\ \mathbf{K}_{t+1} = \frac{\mathbf{F} \mathbf{P}_t \mathbf{H}^T}{R_2 + \mathbf{H} \mathbf{P}_t \mathbf{H}^T} \\ \mathbf{P}_{t+1} = \mathbf{F} \mathbf{P}_t \mathbf{F}^T - \frac{\mathbf{F} \mathbf{P}_t \mathbf{H}^T \mathbf{H} \mathbf{P}_t \mathbf{F}^T}{R_2 + \mathbf{H} \mathbf{P}_t \mathbf{H}^T} + R_1 \end{cases}$$

Recursive Least Squares. model:

$$\begin{cases} \theta_{t+1} = \theta_t + v_t \\ y_t = \varphi_t^T \theta_t + e_t \end{cases}$$

then

$$\begin{cases} \hat{\theta}_t = \hat{\theta}_{t-1} + \mathbf{K}_t \left(y_t - \varphi_t^T \hat{\theta}_{t-1} \right) \\ \mathbf{K}_t = \frac{\mathbf{P}_{t-1} \varphi_t^T}{R_2 + \varphi_t^T \mathbf{P}_{t-1} \varphi_t} \\ \mathbf{P}_t = \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1} \varphi_t \varphi_t^T \mathbf{P}_{t-1}^T}{R_2 + \varphi_t^T \mathbf{P}_{t-1} \varphi_t} + R_1 \end{cases}$$

Let $R_2 = 1$ for simplicity. The tracking capacity is characterized by the covariance matrix $R_1 \in \mathbb{R}^{n \times n}$.

- View R_1 as a design variable.
- Let R_1 be a diagonal matrix.
- Large elements of R_1 imply large parameter variations, and vice versa.
- The Kalman filter gives higher flexibility than the weighted RLS.

Common Problems for Recursive Identification

- Lack of PE.
- Estimator windup.
- $\mathbf{P}_t$ becoming indefinite.

Excitation

Just as for the batch-case, it is important that the input is PE of sufficiently high order. This applies during the whole identification period.

Estimator Windup

Often, some periods of the identification experiment exhibit poor excitation. This causes problems for the identification algorithms. Consider the situation where $\varphi_t = 0$ in the RLS algorithm, then

$$\begin{cases} \hat{\theta}_t = \hat{\theta}_{t-1} \\ \mathbf{P}_t = \frac{1}{\lambda}\mathbf{P}_{t-1} \end{cases}$$

- Notice that $\hat{\theta}$ remains constant during this period,
- ... and $\mathbf{P}$ increases exponentially with time when $\lambda < 1$.

When the system is excited again ($\varphi_t \neq 0$), the estimation gain $\mathbf{K}$ will be very large, and there will be an abrupt change in the estimate, despite the fact that the system has not changed. This effect is referred to as 'estimator windup'.

$\mathbf{P}_t$ becoming indefinite.

$\mathbf{P}_t$ are covariance matrices, so they must be symmetric and positive definite (invertible). Rounding errors however may accumulate and make $\mathbf{P}_t$, and the algorithm may come in trouble. A solution is based on the observation that any positive matrix can be written as the product of two arbitrary matrices (here $\mathbf{S}_t$), or

$$\mathbf{P}_t = \mathbf{S}_t \mathbf{S}_t^T$$

One can then rederive the algorithm in terms of the recursion on such an $\mathbf{S}_t$ (Potter's square root algorithm)

Conclusions

- In practical scenarios, one often needs recursive identification (time-varying, online identification, fault detection).
- Both the OLS and IVM can be cast in recursive forms, the PEM can only be approximated by a recursive algorithm.
- The properties of the online methods are comparable to the offline case.
- Tracking capabilities can be incorporated by using a forgetting factor, or by modeling the parameter variations explicitly.
- Trade-offs between convergence speed and tracking properties, as well as between computational complexity and accuracy.
- In practice, one uses simplifications to make the recursion (i) cheaper, and (ii) more robust.