UPPSALA UNIVERSITET

Informationsteknologi

Introduction to recursive identification algorithms

Torsten Söderström

Systems and Control
Dept of Information Technology
Uppsala University, Sweden

UPPSALA UNIVERSITET

Informationsteknologi

Contents

UPPSALA UNIVERSITET

Informationsteknologi

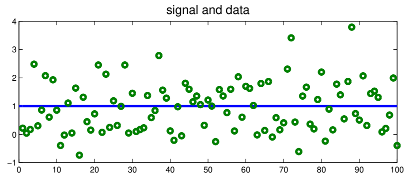
Introduction

Consider the problem of estimating a constant signal from noisy measurements.
Model

$$y(t) = s + e(t)$$

Natural estimate

$$\hat{s}(t) = \frac{1}{t} \sum_{s=1}^{t} y(s)$$



UPPSALA UNIVERSITET

Informationsteknologi

Making the estimate recursive

Aim: Rewrite the estimate so that $\hat{s}(t)$ is updated from $\hat{s}(t-1)$ when the new measurement $y(t)$ is available.
Solution:

$$\begin{aligned}
\hat{s}(t) &= \frac{1}{t}\left[y(t) + \sum_{s=1}^{t-1} y(s)\right] \\
&= \frac{1}{t}\left[y(t) + (t-1)\hat{s}(t-1)\right] \\
&= \hat{s}(t-1) + \frac{1}{t}\left[y(t) - \hat{s}(t-1)\right]
\end{aligned}$$

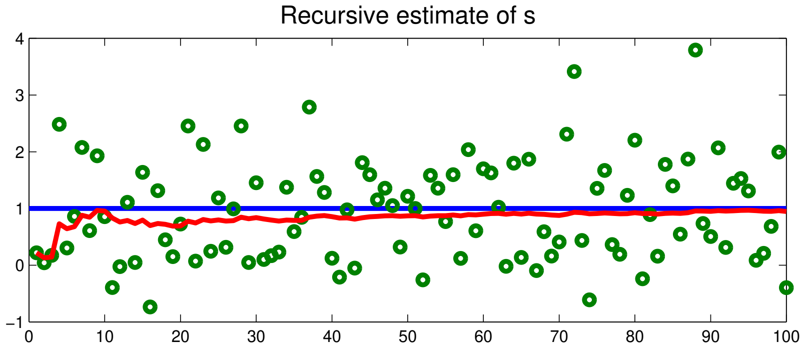
This equation captures the essence of recursive algorithms!

Interpretation

$$\hat{s}(t) = \hat{s}(t-1) + \frac{1}{t}\left[y(t) - \hat{s}(t-1)\right]$$

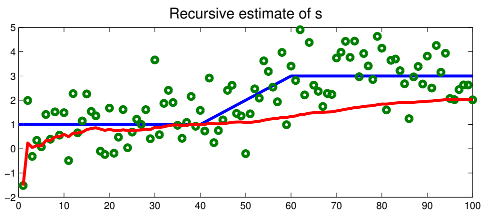
$\hat{s}(t)$	new estimate
$\hat{s}(t-1)$	old estimate
$1/t$	gain sequence
$y(t)$	new measurement
$\hat{s}(t-1)$	prediction of $y(t)$ based on available data at time $t-1$ and model parameter $\hat{s}(t-1)$
$y(t) - \hat{s}(t-1)$	prediction error

Simulations



Time-varying signals

Assume $s(t)$ changes with time.



After a change, give less weight to old data.
How?

Approaches for time-varying case

- Time-invariant case
  $V_t(\theta) = \sum_{s=1}^{t}[y(s) - \theta]^2$
- Exponential forgetting
  $V_t^{(e)}(\theta) = \sum_{s=1}^{t} \lambda^{t-s}[y(s) - \theta]^2$
- Rectangular forgetting
  $V_t^{(r)}(\theta) = \sum_{s=t-L+1}^{t}[y(s) - \theta]^2$
- Observer
  $\hat{s}(t+1) = \hat{s}(t) + K[y(t) - \hat{s}(t)]$

Exponential forgetting

Criterion

$$\begin{aligned} V_t^{(e)}(\theta) &= \sum_{s=1}^{t} \lambda^{t-s}[y(s)-\theta]^2 \\ &= \theta^2 \sum_{s=1}^{t} \lambda^{t-s} - 2\theta \sum_{s=1}^{t} \lambda^{t-s} y(s) + \text{const.} \end{aligned}$$

Estimate

$$\begin{aligned} \hat{\theta}(t) &= \frac{\sum_{s=1}^{t} \lambda^{t-s} y(s)}{\sum_{s=1}^{t} \lambda^{t-s}} = \dots \\ &= \hat{\theta}(t-1) + \frac{1-\lambda}{1-\lambda^t}[y(t)-\hat{\theta}(t-1)] \end{aligned}$$

Rectangular forgetting

Criterion

$$\begin{aligned} V_t^{(r)}(\theta) &= \sum_{s=t-L+1}^{t} [y(s)-\theta]^2 \\ &= \theta^2 \times L - 2\theta \sum_{s=t-L+1}^{t} y(s) + \text{const.} \end{aligned}$$

Estimate

$$\begin{aligned} \hat{\theta}(t) &= \frac{1}{L} \sum_{s=t-L+1}^{t} y(s) \\ &= \hat{\theta}(t-1) + \frac{1}{L}[y(t)-y(t-L)] \end{aligned}$$

Observer-based estimator

Model

$$\begin{aligned} s(t+1) &= s(t) \\ y(t) &= s(t) + e(t) \end{aligned}$$

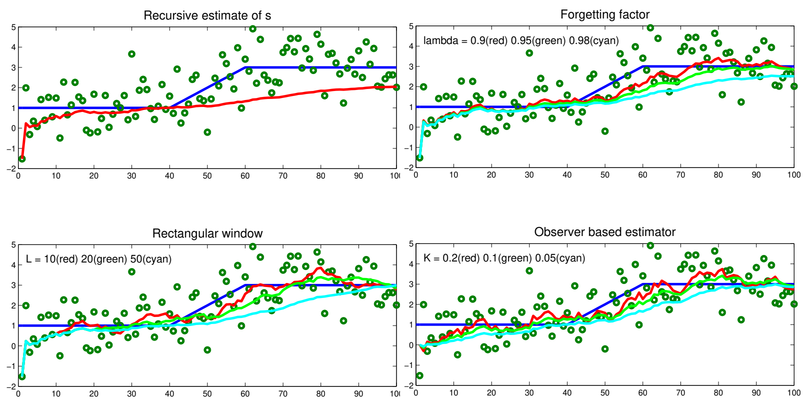
Observer

$$\hat{s}(t+1) = \hat{s}(t) + K[y(t) - \hat{s}(t)]$$

Observer error: $\tilde{s}(t) = s(t) - \hat{s}(t)$

$$\tilde{s}(t+1) = (1-K)\tilde{s}(t) + Ke(t)$$

Simulations



UPPSALA UNIVERSITET

Informationsteknologi

Noise sensitivity

1. Time-invariant case: $E\tilde{s}^2(t) = \sigma^2/t$
2. Time-varying case (when $t \to \infty$):
   - Exponential forgetting: $E\tilde{s}^2(t) = \sigma^2 \frac{1-\lambda}{1+\lambda}$
   - Rectangular forgetting: $E\tilde{s}^2(t) = \sigma^2/L$
   - Observer: $E\tilde{s}^2(t) = \sigma^2 \frac{K}{2-K}$

UPPSALA UNIVERSITET

Informationsteknologi

Inherent tradeoff

- Decrease noise sensitivity
  $[\lambda \approx 1$; L large; $K \approx 0]$

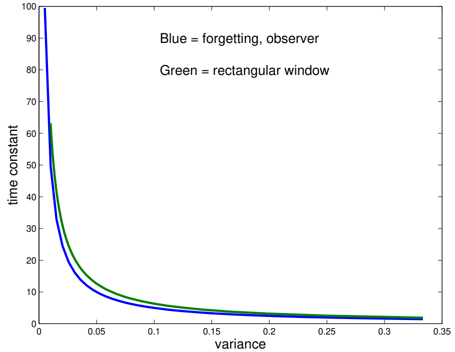
- Fast tracking capability
  $[\lambda \ll 1$; L small; $K \approx 1]$

UPPSALA UNIVERSITET

Informationsteknologi

Inherent tradeoff, cont'd

Time constant for step change in $s(t)$ vs estimation error variance



UPPSALA UNIVERSITET

Informationsteknologi

Approaches for recursive identification

- Modification of off-line methods
- Nonlinear filtering
- Stochastic approximation
- Pseudolinear regression and model reference techniques

UPPSALA UNIVERSITET

Informationsteknologi

Approach 1: Modify off-line algorithms

Least squares method, linear regression model

$$y(t) = \varphi^T(t)\theta + v(t)$$

The regressor vector $\varphi(t)$ depends on data available at time t.

Least squares estimate at time t:

$$\left[\sum_{s=1}^{t} \varphi(s)\varphi^T(s)\right]\hat{\theta}(t) = \left[\sum_{s=1}^{t} \varphi(s)y(s)\right]$$

UPPSALA UNIVERSITET

Informationsteknologi

Derivation of RLS

Set

$$R(t) = \sum_{s=1}^{t} \varphi(s)\varphi^T(s), \quad P(t) = R^{-1}(t)$$

$$\begin{aligned} R(t) &= R(t-1) + \varphi(t)\varphi^T(t) \\ &\Rightarrow \text{[matrix inversion lemma]} \\ P(t) &= P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1+\varphi^T(t)P(t-1)\varphi(t)} \end{aligned}$$

UPPSALA UNIVERSITET

Informationsteknologi

Updating the parameter estimates

$$\begin{aligned} \hat{\theta}(t) &= P(t)\left[\varphi(t)y(t) + \sum_{s=1}^{t-1} \varphi(s)y(s)\right] \\ &= P(t)\left[\varphi(t)y(t) + \left(R(t) - \varphi(t)\varphi^T(t)\right)\hat{\theta}(t-1)\right] \\ &= \hat{\theta}(t-1) + P(t)\varphi(t)\left[y(t) - \varphi^T(t)\hat{\theta}(t-1)\right] \\ &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \end{aligned}$$

Prediction error

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

UPPSALA UNIVERSITET

Informationsteknologi

Estimator gain

$$\begin{aligned} K(t) &= P(t)\varphi(t) \\ &= P(t-1)\varphi(t) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{1+\varphi^T(t)P(t-1)\varphi(t)}\varphi(t) \\ &= P(t-1)\varphi(t)\left(1 - \frac{\varphi^T(t)P(t-1)\varphi(t)}{1+\varphi^T(t)P(t-1)\varphi(t)}\right) \\ &= P(t-1)\varphi(t)\frac{1}{1+\varphi^T(t)P(t-1)\varphi(t)} \end{aligned}$$

UPPSALA UNIVERSITET

Informationsteknologi

Comments on RLS

- No approximations here!
- An instrumental variable estimator can be written in a similar way [Substitute $\varphi(t)$ by $z(t)$; keep $\varphi^T(t)$].
- Recursive (or on-line) algorithms RLS and RIV have the same convergence properties ($t \to \infty$) as the batch (or off-line) algorithms.
- How to set initial values $P(0),\ \hat{\theta}(0)$?

UPPSALA UNIVERSITET

Informationsteknologi

Recursive prediction error method

Prediction error criterion

$$V_t(\theta) = \frac{1}{2}\sum_{s=1}^{t} \varepsilon^2(s,\theta)$$

The model structure specifies how the predition error $\varepsilon(t,\theta)$ is to be computed from available input and output data.
An exact recursive minimizer of $V_t(\theta)$ does not exist in general.
Important exception: LS, for linear regression model.
In the general case: Make approximations, that hold without errors for the LS case.

UPPSALA UNIVERSITET

Informationsteknologi

RPEM, cont'd

Taylor series expansion

$$\begin{aligned} V_t(\theta) &= V_t(\hat{\theta}(t-1)) + V_t'(\hat{\theta}(t-1))(\theta - \hat{\theta}(t-1)) \\ &+ \frac{1}{2}(\theta - \hat{\theta}(t-1))^T V_t''(\hat{\theta}(t-1))(\theta - \hat{\theta}(t-1)) \end{aligned}$$

Minimize!

$$\hat{\theta}(t) = \hat{\theta}(t-1) - \left[V_t''(\hat{\theta}(t-1))\right]^{-1}\left[V_t'(\hat{\theta}(t-1))\right]^T$$

UPPSALA UNIVERSITET

Informationsteknologi

RPEM, cont'd

Negative gradient

$$\psi(t,\theta) = \left[-\frac{\partial}{\partial\theta}\varepsilon(t,\theta)\right]^T$$

Recursion of V_t implies

$$\begin{aligned} V_t'(\theta) &= -\sum \varepsilon(s,\theta)\psi(s,\theta) \\ &= V_{t-1}'(\theta) - \varepsilon(t,\theta)\psi(t,\theta) \\ V_t''(\theta) &= \sum \psi(s,\theta)\psi^T(s,\theta) + \sum \varepsilon(s,\theta)\varepsilon''(s,\theta) \\ &= V_{t-1}''(\theta) + \psi(t,\theta)\psi^T(t,\theta) + \varepsilon(t,\theta)\varepsilon''(t,\theta) \end{aligned}$$

RPEM cont'd

Approximations

- Taylor series of $V_t(\theta)$ up to second order terms
- $V'_{t-1}(\hat{\theta}(t-1)) = 0$
- $\varepsilon\varepsilon''$ neglectable compared to $\psi\psi^T$
- $V''_t(\hat{\theta}(t)) = V''_t(\hat{\theta}(t-1))$

RPEM derivation, cont'd

Set

$$\begin{aligned} R(t) &= \sum_{s=1}^{t} \psi(s,\hat{\theta}(t-1))\psi^T(s,\hat{\theta}(t-1)) \\ &= R(t-1) + \psi(t,\hat{\theta}(t-1))\psi^T(t,\hat{\theta}(t-1)) \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + R^{-1}(t)\psi(t,\hat{\theta}(t-1))\varepsilon(t,\hat{\theta}(t-1)) \end{aligned}$$

How to compute
$\psi(s,\hat{\theta}(t-1)),\quad \varepsilon(s,\hat{\theta}(t-1)),\quad s \le t$?

Computing prediction errors

Find approximation (depends on the model parameterization)

$$\varepsilon(t,\hat{\theta}(t-1)) \sim \varepsilon(t), \quad \psi(t,\hat{\theta}(t-1)) \sim \psi(t)$$

Set $P(t) = R^{-1}(t)$. Algorithm

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\ P(t) &= P(t-1) - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{1+\psi^T(t)P(t-1)\psi(t)} \\ K(t) &= P(t)\psi(t) = \frac{P(t-1)\psi(t)}{1+\psi^T(t)P(t-1)\psi(t)} \end{aligned}$$

Approximate prediction errors

Principle: Use recent parameter estimates
Example: MA(1) process

$$\begin{aligned} y(t) &= e(t) + ce(t-1), \quad \theta = c \\ \varepsilon(t,\theta) &= \frac{1}{1+cq^{-1}} y(t) \\ \psi(t,\theta) &= \frac{-1}{(1+cq^{-1})^2} y(t-1) = \frac{-1}{1+cq^{-1}} \varepsilon(t-1,\theta) \end{aligned}$$

Approximation

$$\begin{aligned} \varepsilon(t) &= -\hat{c}(t-1)\varepsilon(t-1) + y(t) \\ \psi(t) &= -\hat{c}(t-1)\psi(t-1) - \varepsilon(t-1) \end{aligned}$$

Approach 2: Nonlinear filtering

Regard θ as a random variable.

Find the conditional pdf $p(\theta|Z^t)$,
$Z^t = \{y(1), u(1), \dots y(t), u(t)\}$.

Choose

$$\hat{\theta}(t) = E(\theta|Z^t)$$

Nonlinear filtering, linear regresssion

Model

$$y(t) = \varphi^T(t)\theta + e(t), \quad e(t) \sim N(0, r_2)$$

Here, $\varphi(t)$ depends on Z^{t-1}.
A priori distribution $\theta \sim N(\theta_o, P_o)$.

Use Bayes' formula

$$\begin{aligned} p(\theta|Y^t) &= p(\theta|Y^{t-1}, y(t)) = \frac{p(y(t)|\theta, Y^{t-1})p(\theta|Y^{t-1})}{p(y(t)|Y^{t-1})} \\ p(\theta|Y^0) &\sim N(\theta_o, P_o) \end{aligned}$$

Nonlinear filtering, cont'd

$$p(y(t)|\theta, Y^{t-1}) \sim N(\varphi^T(t)\theta, r_2(t))$$

Assume

$$p(\theta|Y^{t-1}) \sim N(\hat{\theta}(t-1)|P(t-1))$$

Use induction to show

$$p(\theta|Y^t) \sim N(\hat{\theta}(t)|P(t))$$

How are $\hat{\theta}(t)$ and $P(t)$ updated?

Nonlinear filtering algorithm

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\ K(t) &= \frac{P(t-1)\varphi(t)}{r_2 + \varphi^T(t)P(t-1)\varphi(t)} \\ P(t) &= P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{r_2 + \varphi^T(t)P(t-1)\varphi(t)} \end{aligned}$$

Kalman filter interpretation

Linear regression model

$$\begin{aligned}\theta(t+1) &= \theta(t) \quad [= x] \\ y(t) &= \varphi^T(t)\theta + e(t)\end{aligned}$$

Find

$$\hat{x}(t+1) = E[x(t+1)|Y^t] = \hat{x}(t+1|t)$$

Kalman filter, cont'd

$$\begin{aligned}\hat{x}(t+1) &= \hat{x}(t) + K(t)[y(t) - \varphi^T(t)\hat{x}(t-1)] \\ K(t) &= \frac{\overline{P}(t)\varphi(t)}{r_2 + \varphi^T(t)\overline{P}(t)\varphi(t)} \\ \overline{P}(t) &= \overline{P}(t-1) + R_1 - \frac{\overline{P}(t)\varphi(t)\varphi^T(t)\overline{P}(t)}{r_2 + \varphi^T(t)\overline{P}(t)\varphi(t)}\end{aligned}$$

(Set $\overline{P}(t) = P(t-1)$.)

Note: Interpretation of $\hat{\theta}(0), P(0)$.

Nonlinear filtering, general case

State-space model

$$\begin{aligned}x(t+1) &= F(\theta)x(t) + G(\theta)u(t) + w(t) \\ y(t) &= H(\theta)x(t) + e(t) \\ &\quad \operatorname{cov}(w(t)) = R_1(\theta),\ \operatorname{cov}(e(t)) = R_2(\theta)\end{aligned}$$

Form an extended state space model using

$$\overline{x}(t) = \begin{pmatrix} x(t) \\ \theta \end{pmatrix}$$

and use, say, an extended Kalman filter for estimating $\overline{x}(t)$.

Approach 3: Stochastic approximation

Problem: Solve

$$0 = f(x) = EQ(x, e(t))$$

w r t x; $e(t)$ is unknown; $Q(x, e)$ can be observed for arbitrary x.

Example: Minimize

$$V(\theta) = \frac{1}{2}E[y(t) - \varphi^T(t)\theta]^2$$

$$0 = E\varphi(t)[y(t) - \varphi^T(t)\theta]$$

Here, $x = \theta, \quad e(t) = \begin{pmatrix} y(t) & \varphi(t) \end{pmatrix}$.

Stochastic approximation algorithm

$$\hat{x}(t) = \hat{x}(t-1) + \gamma(t)Q(\hat{x}(t-1), e(t))$$

Steplength $\gamma(t)$ satisfies

$$\gamma(t) > 0, \quad \gamma(t) \to 0, t \to \infty$$

Usual conditions

$$\sum \gamma(t) = \infty, \quad \sum \gamma^2(t) < \infty$$

Example: $\gamma(t) = \frac{1}{t}$

Stochastic approximation example

Solve

$$E[e(t) - x] = 0$$

[Hence, $x = Ee(t)$.] Algorithm with $\gamma(t) = 1/t$:

$$\begin{aligned}
\hat{x}(t) &= \hat{x}(t-1) + \frac{1}{t}[e(t) - \hat{x}(t-1)] \\
&= \frac{t-1}{t}\hat{x}(t-1) + \frac{1}{t}e(t) \\
t\hat{x}(t) &= (t-1)\hat{x}(t-1) + e(t) \\
t\hat{x}(t) &= \sum_{s=1}^{t} e(s) \Rightarrow \hat{x}(t) = \frac{1}{t}\sum_{s=1}^{t} e(s)
\end{aligned}$$

SA algorithm, linear regresion

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)]$$

LMS (least mean squares)

Some gradient algorithms:

- $\gamma(t) = \gamma_o \quad (\text{fix})$
- $\gamma(t) = \frac{\gamma_o}{|\varphi(t)|^2} \quad (\text{normed})$
- $\gamma(t) = \frac{\gamma_o}{\sum_{s=1}^{t} |\varphi(s)|^2}$

SA algorithm, Newton direction

Set

$$V'' = E\varphi(t)\varphi^T(t)$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)\widehat{V''}^{-1}Q(\hat{\theta}(t-1), e(t))$$

Update $V'' \Rightarrow$
Algorithm

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + \gamma(t)R^{-1}(t)\varphi(t)[y(t) - \varphi^T(t)\hat{\theta}(t-1)] \\
R(t) &= R(t-1) + \gamma(t)[\varphi(t)\varphi^T(t) - R(t-1)]
\end{aligned}$$

Approach 4: Pseudolinear regression

Idea: Treat $\varepsilon(t)$ as a second input, known at times $s \geq t+1$. Apply standard RLS.

The idea is applicable to the general linear model

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

(and special cases thereof).

UPPSALA UNIVERSITET
Informationsteknologi

PLR for ARMA models

Model

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

$$\begin{aligned} y(t) &= \varphi_o^T(t)\theta + e(t) \\ \theta &= \begin{pmatrix} a_1 & \dots & c_n \end{pmatrix}^T \\ \varphi_o^T(t) &= \begin{pmatrix} -y(t-1) & \dots & e(t-n) \end{pmatrix} \end{aligned}$$

Define model regressor

$$\varphi(t) = \begin{pmatrix} -y(t-1) & \dots & \varepsilon(t-n) \end{pmatrix}$$

UPPSALA UNIVERSITET
Informationsteknologi

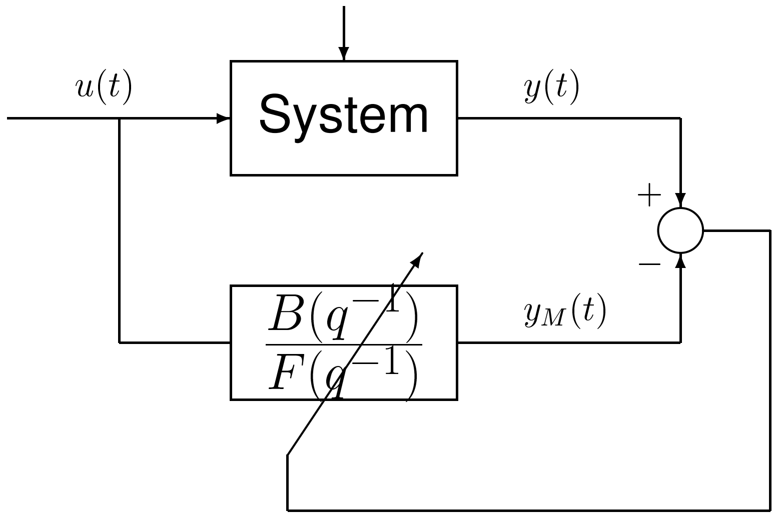
PLR algorithm

Note $\varphi(t)$ is formed from Y^{t-1} and $\hat{\theta}(t-1),\ \hat{\theta}(t-2), \dots$

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

UPPSALA UNIVERSITET
Informationsteknologi

Model reference technique



Model reference technique, cont'd

Find unknown output $\hat{y}_M(t)$.

$$y(t) \approx \frac{B(q^{-1})}{F(q^{-1})} u(t)$$

Rewrite model

$$\begin{aligned} \hat{y}_M(t) &= \varphi^T(t)\hat{\theta}(t-1) \\ &= \begin{pmatrix} -\hat{y}_M(t-1) & \ldots & u(t-n) \end{pmatrix} \begin{pmatrix} \hat{f}_1(t-1) \\ \vdots \\ \hat{b}_n(t-1) \end{pmatrix} \end{aligned}$$

Model reference algorithm

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + P(t)\varphi(t)\left[y(t) - \hat{y}_M(t)\right] \\ P(t) &= \left[\sum_{s=1}^{t} \varphi(s)\varphi^T(s)\right]^{-1} \end{aligned}$$

This can be interpreted as a recursive output error method.

The algorithm has better convergence than RLS (often no bias).

Tracking time-varying dynamics

Basic algorithm

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

When the true parameter vector is time-varying, say, $\theta_o(t)$, attempt $\hat{\theta}(t) \approx \theta_o(t)$.

Prevent $K(t)$ from decreasing to 0: For example,

- Modify criterion
- Modify gain sequence $\gamma(t)$
- Modify model

Tracking, Bayes' approach

Modified model

$$\begin{aligned} \theta(t+1) &= \theta(t) + v(t), \quad \mathrm{cov}(v(t)) = R_1 \\ y(t) &= \varphi^T(t)\theta(t) + e(t), \quad \mathrm{cov}(e(t)) = R_2 \end{aligned}$$

Algorithm

$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\ \varepsilon(t) &= y(t) - \varphi^T(t)\hat{\theta}(t-1) \\ K(t) &= \frac{P(t-1)\varphi(t)}{r_2 + \varphi^T(t)P(t-1)\varphi(t)} \\ P(t) &= P(t-1) + R_1 - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{r_2 + \varphi^T(t)P(t-1)\varphi(t)} \end{aligned}$$

Tracking, modified off-line approach

Modified criterion

$$V_t(\theta) = \frac{1}{2}\sum_{s=1}^{t} \lambda^{t-s}\varepsilon^2(s,\theta)$$

with forgetting factor $\lambda < 1$. Algorithm

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

Using a forgetting factor, cont'd

Modified criterion

$$V_t(\theta) = \frac{1}{2}\sum_{s=1}^{t} \lambda^{t-s}\varepsilon^2(s,\theta)$$

Assume $\lambda \approx 1$.
Memory, time constant T?

$$\lambda^T = e^{T\log(\lambda)} \approx e^{T(\lambda-1)} = e^{-1}$$

gives

$$T = \frac{1}{1-\lambda}$$

Tracking, SA algorithm

Choose gain sequence such that $\gamma(t) \not\to 0$.

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + \gamma(t)R^{-1}(t)\varphi(t)\varepsilon(t)\\
R(t) &= R(t-1) + \gamma(t)[\varphi(t)\varphi^T(t) - R(t-1)]
\end{aligned}$$

Compare with

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)\\
P(t) &= \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\right]\frac{1}{\lambda(t)}
\end{aligned}$$

Are the two algorithms identical?

Tracking, SA algorithms

The two algorithms (SA-based, and RPEM based, resp.) are identical, if

$$\lambda(t) = \frac{1-\gamma(t)}{\gamma(t)}\gamma(t-1)$$

Examples:

1. $\gamma(t) = \gamma_o \Rightarrow \lambda(t) = \lambda = 1 - \gamma_o$
2. $\gamma(t) = \frac{1}{t} \Rightarrow \lambda(t) = 1$

Informationsteknologi

UPPSALA UNIVERSITET

Summary

There is only one recursive identification algorithm,

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

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

(+ a part due to the model structure)

A large number of varieties, due to various user choices.

Informationsteknologi

UPPSALA UNIVERSITET

User choices in recursive algorithms

- Model structure/model parameterization
- Input signal [experimental condition]
- Criterion function
- Gain sequence $\gamma(t)$ [or forgetting factor $\lambda(t)$]
- Search direction [gradient vs. Newton]
- Initial conditions [$\theta(0),\ P(0)$]
- $[$Optional: instrumental variable variant$]$
- $[$Optional: PLR variant$]$