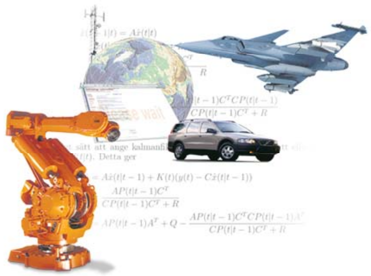
Digital Signal Processing, Lecture 12
Design of Adaptive Algorithms and Examples



Thomas Schön
Division of Automatic Control
Department of Electrical Engineering
Linköping University

E-mail: schon@isy.liu.se
Phone: 281373
Office: House B, entrance 27

Outline Lecture 12

1. Summary of lecture 11
2. Design considerations
3. Example
4. Estimation error for adaptive algorithms
5. State estimation revisited
6. A few words about the particle filter

Summary of Lecture 11 (I/III)

Adaptive signal processing algorithms deals with the problem of estimating the time-varying parameters in real-time, where

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

The algorithms are in the form

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right)$$

Summary of Lecture 11 (II/III)

Recursive Least Squares (RLS)

The following cost function is minimized (LS with a forgetting factor)

$$V_t(\theta) = \sum_{k=1}^{t} \lambda^{t-k}(y(k) - \varphi^T(k)\theta)^2$$

Forgetting factor, the only difference to the least squares algorithm.

The recursive solution is given by

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

Summary of Lecture 11 (III/III)

Least Mean Square (LMS)

The following cost function is minimized

$$V_t(\theta) = E\left(y(k) - \varphi^T(k)\theta\right)^2$$

The stochastic gradient method results in the following recursion

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right),$$

Standard (un-normalized) LMS makes use of $K(t) = \mu\varphi(t)$

and the normalized LMS (NLMS) makes us of (α is a small number to avoid division by zero)

$$K(t) = \mu\frac{\varphi(t)}{\alpha + |\varphi(t)|^2}$$

Design Considerations

- Transient error: Increase P_0
- Large variance in the estimates (variance error): Decrease the filter gain
- Slow adaptation (adaption error): Increase the filter gain
- The estimates do not converge (bias error): Increase the model order

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \underbrace{K(t)}_{\text{Filter gain}}\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right)$$

Algorithm Summary

The algorithms are in the form

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\left(y(t) - \varphi^T(t)\hat{\theta}(t-1)\right),$$

where

LMS	$K(t) = \mu\varphi(t)$
NLMS	$K(t) = \frac{\mu\varphi(t)}{\alpha + \varphi^T(t)\varphi(t)}$,
RLS	$K(t) = P(t)\varphi(t)$, $P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$.
KF	$K(t) = \frac{P(t-1)\varphi(t)}{R(t) + \varphi^T(t)P(t-1)\varphi(t)}$, $P(t) = P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{R(t) + \varphi^T(t)P(t-1)\varphi(t)} + Q(t)$.

Ex. Question

Apply the two algorithms to Gaussian white noise with variance 4 (`y=2*randn(10000,1)`).

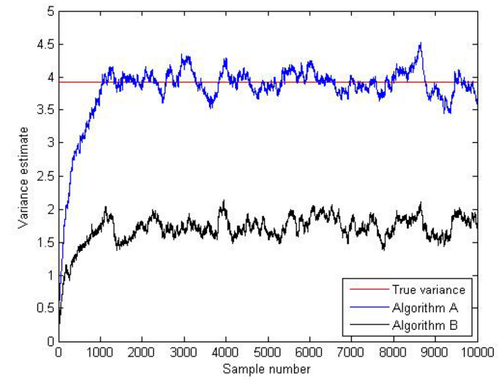
• One of the algorithms converge towards the true value (i.e., no bias error). Select the step length so that the variance of the estimation error is less than 0.04 (i.e., 1% of the correct value 4) when the transient is gone. What is the length of the transient?

• One of the algorithms does not converge to the true value (i.e., bias error). What is the bias error for this algorithm?

Ex. MATLAB Code

```
N = 10000;    % Number of samples
y = 2*randn(N,1);
muA=0.003;    % Step length Alg. A
muB=0.015;    % Step length Alg. B
thA_hat=0;    % Estimate using Alg. A
thB_hat=0;    % Estimate using Alg. B
for k=1:N
  thA_hat     = thA_hat + muA*(y(k)^2-thA_hat);
  thA_vect(k) = thA_hat;
  thB_hat     = thB_hat + muB*sign(y(k)^2-thB_hat);
  thB_vect(k) = thB_hat;
end;
plot(1:N,repmat(var(y),1,N),'r')
hold on;
plot(1:N,thA_vect,'b');
plot(1:N,thB_vect,'k');
hold off;
VarA = var(thA_vect(1500:end))
ylabel('Variance estimate')
xlabel('Sample number')
legend('True variance','Algorithm A','Algorithm B')
```


Ex. Result



Algorithm A converge to the true value and Algorithm B does not converge to the true value. The length of the transient is roughly 1200 samples. The bias error of Alg. B is 4 – 1.8 = 2.2.

Ex. Estimation Error – LMS

In LMS, the filter gain is given by

$$K(t) = \underbrace{\mu}_{\text{step length}}\underbrace{\varphi(t)}_{\text{regression vector}}$$

Assume (for simplicity) that θ is scalar, which implies that the regression vector is scalar.

→ Stability requires: $|1 - \mu\varphi^2(t)| < 1$

Hence, choose

$$0 < \mu < \frac{2}{\varphi^2(t)}$$

In practise we reduce the upper limit by a factor of 100, i.e., $\mu = \dfrac{0.02}{\varphi^2(t)}$

Recall – the Kalman Filter

Kalman filter problem definition: Compute an estimate of the state vector x(t) for a state-space model

$$x(t+1) = Ax(t) + w(t),$$
$$y(t) = Cx(t) + v(t),$$

using the information available in all the previous measurements

$$y(\tau), \qquad 0 \leq \tau \leq t$$

such that the covariance of the estimation error

$$\tilde{x}(t \mid t) = x(t) - \hat{x}(t \mid t)$$

is minimized.

Nonlinear State Estimation – the Model

Nonlinear state-space model (stochastic difference equation)

State variable

Input signal

Process model $x_{t+1} = f_t(x_t, u_t) + v_t$ ← Stochastic process noise

Measurement model $y_t = h_t(x_t, u_t) + e_t$ ← Stochastic measurement noise

Measurement

Alternative formulation
$$x_{t+1} \sim p(x_{t+1}|x_t)$$
$$y_t \sim p(y_t|x_t)$$

The two model descriptions are related,
$$p(x_{t+1}|x_t) = p_{v_t}(x_{t+1} - f_t(x_t))$$
$$p(y_t|x_t) = p_{e_t}(y_t - h_t(x_t))$$

Nonlinear State Estimation – the Aim

The aim is to obtain information about the state x_t using the information present in the measurements $y_{1:t} = \{y_i\}_{i=1}^t$

Compute $p(x_t|y_{1:t})$

State Estimation – Conceptual Solution

We have now shown that for the dynamic model
$$x_{t+1} \sim p(x_{t+1}|x_t)$$
$$y_t \sim p(y_t|x_t)$$
the filtering and the one-step ahead prediction densities are

Measurement update
$$p(x_t|y_{1:t}) = \frac{\overbrace{p(y_t|x_t)}^{\text{Likelihood}}\overbrace{p(x_t|y_{1:t-1})}^{\text{Previous prediction density}}}{p(y_t|y_{1:t-1})}$$

Time update
$$p(x_{t+1}|y_{1:t}) = \int \underbrace{p(x_{t+1}|x_t)}_{\text{Dynamics}}\underbrace{p(x_t|y_{1:t})}_{\text{Previous filtering density}} dx_t$$

Special Case – Kalman Filter

Important special case, linear equations with Gaussian noise

$$\boxed{\begin{aligned} x_{t+1} &= Ax_t + v_t, & v_t &\sim \mathcal{N}(0, Q), \\ y_t &= Cx_t + e_t, & e_t &\sim \mathcal{N}(0, R) \end{aligned}}$$

This allows for a closed form solution

$$p(x_t|y_{1:t}) = \mathcal{N}(x_t; \hat{x}_{t|t}, P_{t|t}),$$
$$p(x_{t+1}|y_{1:t}) = \mathcal{N}(x_{t+1}; \hat{x}_{t+1|t}, P_{t+1|t}).$$

Gaussian variables and linear transformations implies that complete information is provided by the mean and the covariance.

The Particle Filter – Very Brief Introduction

The particle filter computes estimates of the filtering PDF

The key idea is to form a nonparametric estimate,

$$p^N(x_t|y_{1:t}) = \sum_{i=1}^{N} \underbrace{\boxed{\tilde{q}_t^i}}_{\text{weights}} \delta(x_t - \underbrace{\boxed{x_t^i}}_{\text{particles}}), \qquad \sum_{i=1}^{N} \tilde{q}_t^i = 1, \qquad \tilde{q}_t^i \geq 0, \forall i$$

The estimate converge as $N \to \infty$

This implies that the multidimensional integrals are reduced to finite sums

$$\boxed{\text{“}\ \delta + \int \to \sum\ \text{”}}$$

The Particle Filter in Terms of the Kalman Filter

The Kalman filter

- 1. Initialize
  $x_{0|0} = \bar{x}$,
  $P_{0|0} = P_0$
- 2. Time update
  $\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1}$,
  $P_{t|t-1} = AP_{t-1|t-1}A^T + Q$.
- 3. Measurement update
  $\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - C\hat{x}_{t|t-1})$,
  $P_{t|t} = P_{t|t-1} - K_t C P_{t|t-1}$.

$$\hat{p}(x_t|y_{1:t}) = \mathcal{N}(x_t;\, \hat{x}_{t|t}, P_{t|t})$$

The particle filter

1. Initialize:
   $x_0^i \sim p(x_0)$
2. Time update
   $\bar{x}_t^i = f(x_{t-1}^i) + v_{t-1}^i$
3. Measurement update
   $q_t^i = p(y_t|\bar{x}_t^i) / \sum_{j=1}^{N} p(y_t|\bar{x}_t^j)$,
   $\mathrm{P}(x_t^i = \bar{x}_t^j) = w_t^j$.

$$\hat{p}(x_t|y_{1:t}) = \sum_{i=1}^{N} \frac{1}{N} \delta(x_t - x_t^i)$$

Particle Filter – Illustrating Example

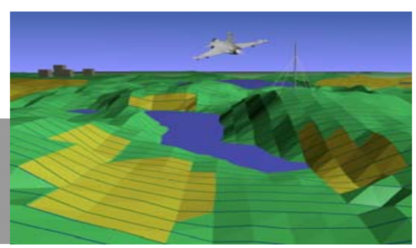
Sensor Fusion

Sensors

- Barometer
- Inertial sensors
- Radar
- Terrain elevation

Estimation problem
- World model
- Dynamic model
- Sensor model

Pose estimates

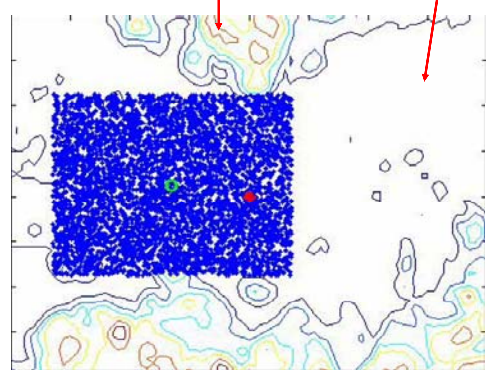


Digital Terrain Elevation Database: 200 000 000 grid points, 50 meter between points, 2.5 meters uncertainty
Ground Cover Database: 14 types of vegetation
Obstacle Database: All man made obstacles above 40 m

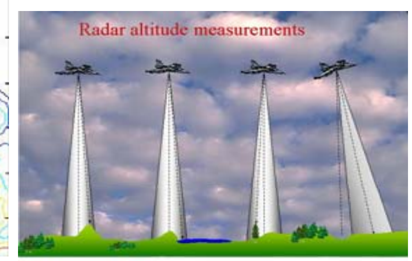
Particle Filter – Illustrating Example

Map with level curves

Water



Green (O) Particle filter estimate
Red (+) True horizontal position
Blue (◊) Particles



A Few Concepts to Summarize Lecture 12

Bias error: The difference between the true parameter and the estimated parameter. If there is a bias error, we say that the estimate is biased.

Bayes' theorem: Explains how conditional random variables depend on each other.

Markov property: Given x(t), there in no useful information present in the previous states and measurements.

Particle filter: A method for estimation the state in a nonlinear dynamic system.

Active Noise Control/Cancelling (ANC): A method for reducing unwanted, disturbing noise.