Summary of lecture 4

I **Signal in noise problem:** $u(t) = s(t) + n(t)$. Remove noise $n(t)$ from the observed signal $u(t)$ to obtain information about $s(t)$.

II **Frequency selective filters:** Amplifies/attenuates frequencies in order to supress noise.

III **Stability and causality:** Poles inside/outside the unit circle $\leftrightarrow$ stable and causal/anti-causal filter.

IV **Linear phase filter:** Time delay is same for all frequencies.

V **Zero phase filter:** No time delay. Non-causal, only used off-line.

VI **Spectral factorization:** Any stationary s.p. $y(t)$ can be realized by filtering white noise $e(t)$ with variance λ through a filter $H(q)$,

$$\Phi_{yy}(\omega) = \lambda \left| H(e^{i\omega}) \right|^2$$

Outline Lecture 5

We have seen that stationary stochastic processes can be represented by

- realizations,
- covariance functions,
- spectra.

All three are of about the same complexity. In this lecture we introduce

parametric signal models,

which is a more compact representation.

I Standard signal sources

II AR, ARMA and state space models

III Converting transfer functions to
   1. spectrum
   2. covariance function
   3. state space model

IV Prediction using signal models

Signal modelling

- Any stationary s.p. $y(t)$ can be represented as

$$y(t) = T(q)u(t)$$

  where $T(q)$ is a filter and $u(t)$ is white noise input signal.
- Super formula in spectral domain gives

$$\Phi_{yy}(\omega) = \left|T\left(e^{i\omega}\right)\right|^2 \Phi_{uu}(\omega) = \left|T\left(e^{i\omega}\right)\right|^2 \lambda$$

- The filter shapes the spectrum, the input injects energy.
- Many $u(t)$ have the property that $\Phi_{yy}(\omega) = \left|T\left(e^{i\omega}\right)\right|^2 \lambda$. This can be used to shape the waveform $y(t)$.

Standard signal sources

A signal $y(t)$ of length N such that $\Phi_{yy}(\omega) = \left|T\left(e^{i\omega}\right)\right|^2 \lambda$, $E_y = N\lambda$.

- White noise. Most common source, often Gaussian distributed

$$u(t) \sim \mathcal{N}(0, \lambda)$$

  Ex: Measurement noise, non-voiced sounds like f, v, s.

- Pulse

$$u(t) = \sqrt{N\lambda}\delta(t) = \begin{cases} \sqrt{N\lambda} & t = 0 \\ 0 & t \neq 0 \end{cases}$$

  Ex: disturbances in a control systems, explosive/implosive sounds like b, d, k, t.

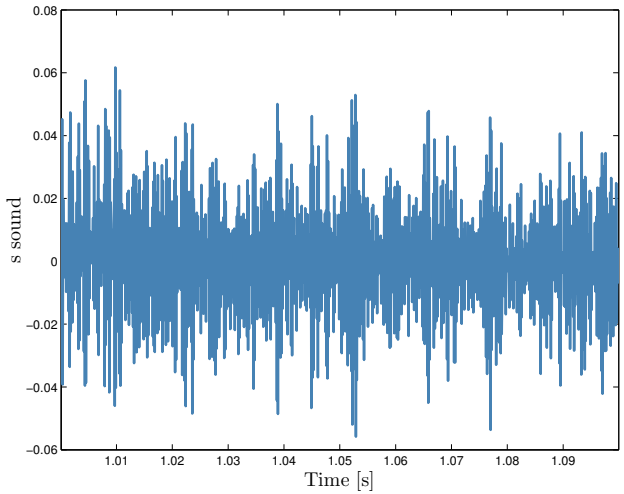
- Pulse train (note difference with book)

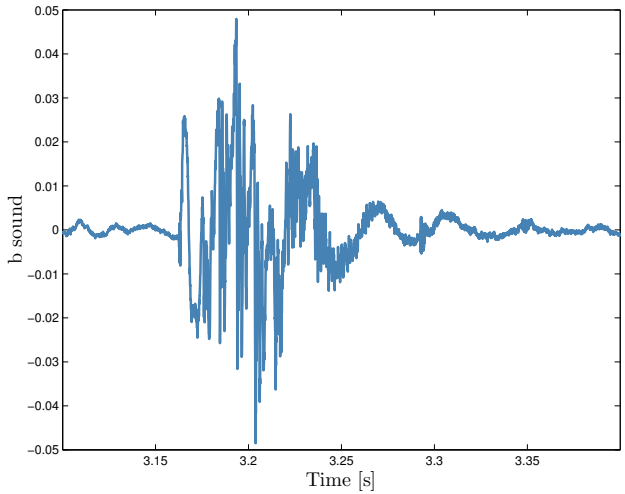
$$u(t) = \sqrt{P\lambda} \sum_{k=0}^{n_P} \delta\left(t - kP\right) \qquad n_P = \left\lceil \frac{N}{P} \right\rceil$$

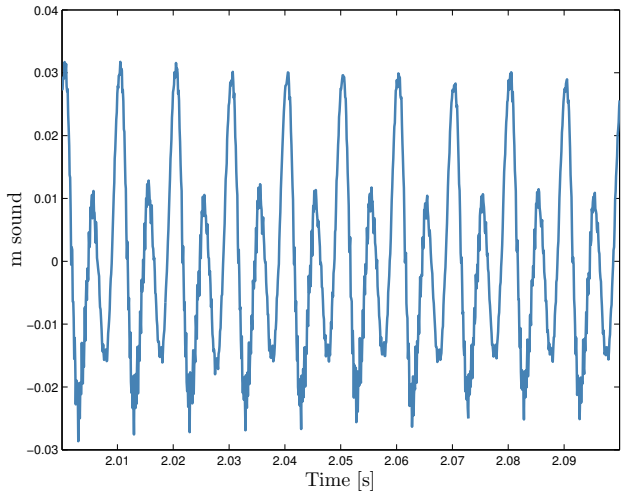
  Ex: voiced sounds like m, n, l.

Standard signal sources: examples

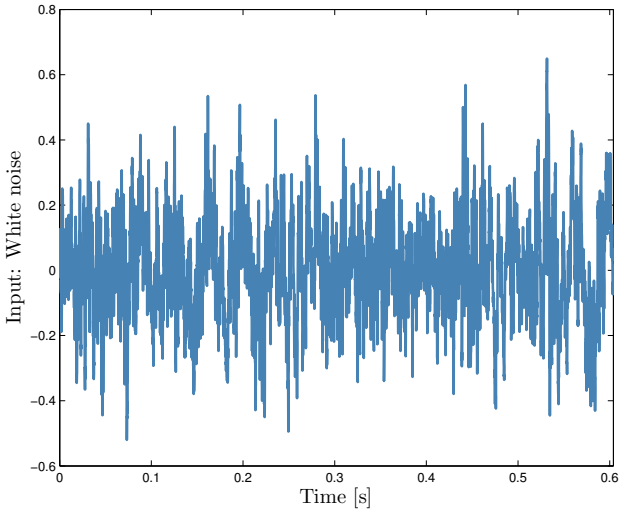
Recordings of s, b and m sound.

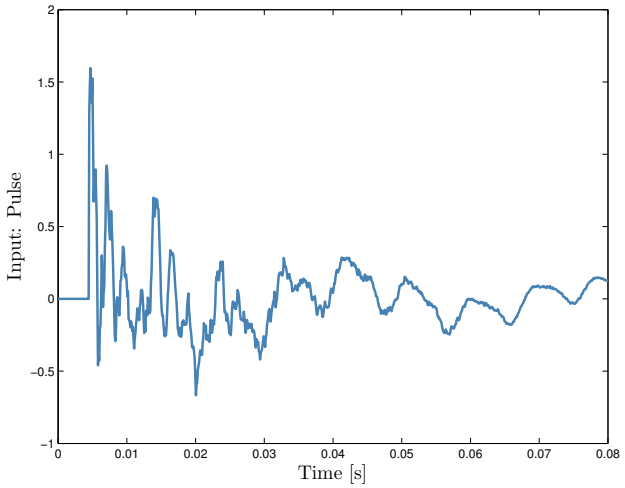


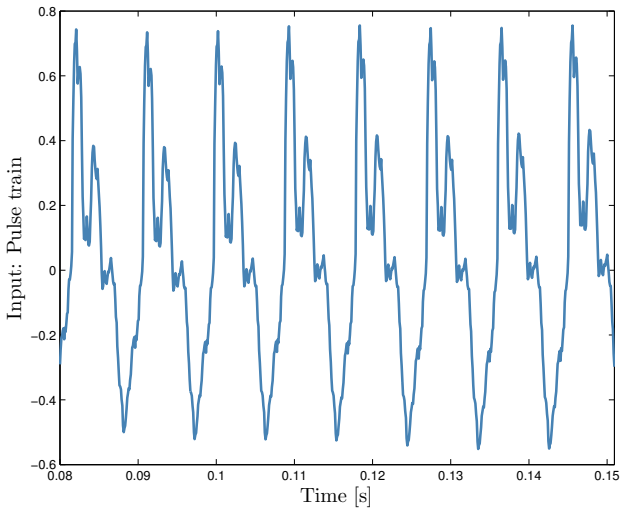




Same filter $T(q)$ for inputs white noise, pulse and pulse train.







State-space models: motivation

There are many good reasons for using state-space models in signal processing, e.g.:

1. Typical result of physical model building, i.e. "easy" to construct and interpret the models.
2. Powerful representation for analysis of the system properties.
3. Used in the Kalman filter
4. Very general model (multivariable, time varying, etc)

Conversion example: AR to state-space

We have an AR model

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = e(t)$$

Letting the state vector be $x(t) = [y(t-1) \ldots y(t-n)]^T$ results in

$$x(t+1) = \begin{bmatrix} -a_1 & -a_2 & \ldots & -a_{n-1} & -a_n \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} e(t)$$

$$y(t) = \begin{bmatrix} -a_1 & -a_2 & \ldots & -a_{n-1} & -a_n \end{bmatrix} x(t) + e(t)$$

Conversion example: ARMA to state-space (I/II)

We have an ARMA model,

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = e(t) + c_1 e(t-1) + \ldots + c_m e(t-m)$$

Let $m = n$ (if not, just add zero coefficients),

$$T(q) = \frac{1 + c_1 q^{-1} + \ldots + c_n q^{-n}}{1 + a_1 q^{-1} + \ldots + a_n q^{-n}} = 1 + \frac{(c_1 - a_1) q^{-1} + \ldots + (c_n - a_n) q^{-n}}{1 + a_1 q^{-1} + \ldots + a_n q^{-n}}$$
$$= 1 + \sum_{k=1}^{n} (c_k - a_k) \frac{q^{-k}}{1 + \ldots + a_n q^{-n}}$$

and introduce the state variables

$$x_1(t) = \frac{q^{-1}}{1 + \ldots + a_n q^{-n}} e(t) \ , \ \ldots \ , \ x_n(t) = \frac{q^{-n}}{1 + \ldots + a_n q^{-n}} e(t)$$

Conversion example: ARMA to state-space (II/II)

Using the relationships

$$x_i(t+1) = qx_i(t) = q\frac{q^{-i}}{A(q)}e(t) = \frac{q^{-(i-1)}}{A(q)}e(t) = x_{i-1}(t)$$

and $q^{-k}x_1(t) = x_{1+k}(t)$ we get

$$x(t+1) = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{n-1} & -a_n \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} e(t)$$

$$y(t) = \begin{bmatrix} c_1 - a_1 & c_2 - a_2 & \dots & c_n - a_n \end{bmatrix} x(t) + e(t)$$

Note that the AR model is just a special case of the above.

Conversion: Matlab

`tf2ss` converts a transfer function $T(q)$ to a state-space model

$$x(t+1) = Ax(t) + Be(t)$$
$$y(t) = Cx(t) + De(t)$$

`ss2tf` converts a state-space model to a transfer function model

$$T(q) = C(qI - A)^{-1}B + D \qquad \text{or} \qquad T(z) = C(zI - A)^{-1}B + D$$

We can always make a variable change $\tilde{x}(t) = Px(t)$, P invertible.
This gives a different state space representation

$$\tilde{x}(t+1) = PAP^{-1}\tilde{x}(t) + PBe(t)$$
$$y(t) = CP^{-1}\tilde{x}(t) + De(t)$$

but the same $T(q)$.

Prediction for transfer function models

We have $y(t+k) = T(q)e(t+k)$ on impulse response form,

$$y(t+k) = \sum_{i=0}^{\infty} t(i)e(t+k-i) = \sum_{i=0}^{k-1} t(i)e(t+k-i) + \sum_{i=k}^{\infty} t(i)e(t+k-i)$$

Replace the unknown future noise signal values with zero in predictor

$$\hat{y}(t+k|t) = \sum_{i=k}^{\infty} t(i)e(t+k-i) = \sum_{j=0}^{\infty} t(j+k)e(t-j) = \tilde{T}_k(q)e(t)$$

$y(t) = T(q)e(t) \Leftrightarrow e(t) = T^{-1}(q)y(t)$, and thus the predictor is

$$\hat{y}(t+k|t) = \tilde{T}_k(q)T^{-1}(q)y(t)$$

Summary of Lecture 5

- **Signal model:** A filter $T(q)$ and a signal source $u(t)$.
- **Signal modelling:** The problem of finding the filter and source that makes the modeled signal as close to the true signal as possible.
- **ARMA model:** Auto-Regressive Moving Average signal model,

$$y(t) + \sum_{k=1}^{n} a_k y(t-k) = e(t) + \sum_{\ell=1}^{m} c_\ell e(t-\ell)$$

- **State space model:** A general and powerful way to model signals. Innovation form,

$$x(t+1) = Ax(t) + Be(t)$$
$$y(t) = Cx(t) + De(t)$$

- **k-step prediction:** Based on information about signal up to time t, use model to compute best prediction of signal value at time $t+k$,

$$\hat{y}(t+k|t)$$