Signal Processing for acoustic neutrino detection

Erice Oct 2013

Sean Danaher
University of Northumbria,
Newcastle UK

Introduction

- You need to Know!
- Fourier
- Laplace
- State Space
- Z-Transform
- SVD
- Conclusions

Sadly a Very Maths Based Subject!

- But Computers Do the Maths e.g. MATLAB, C++ etc
- Need to know conceptually what can be done
- Need to be able to write the problem in the appropriate mathematical format

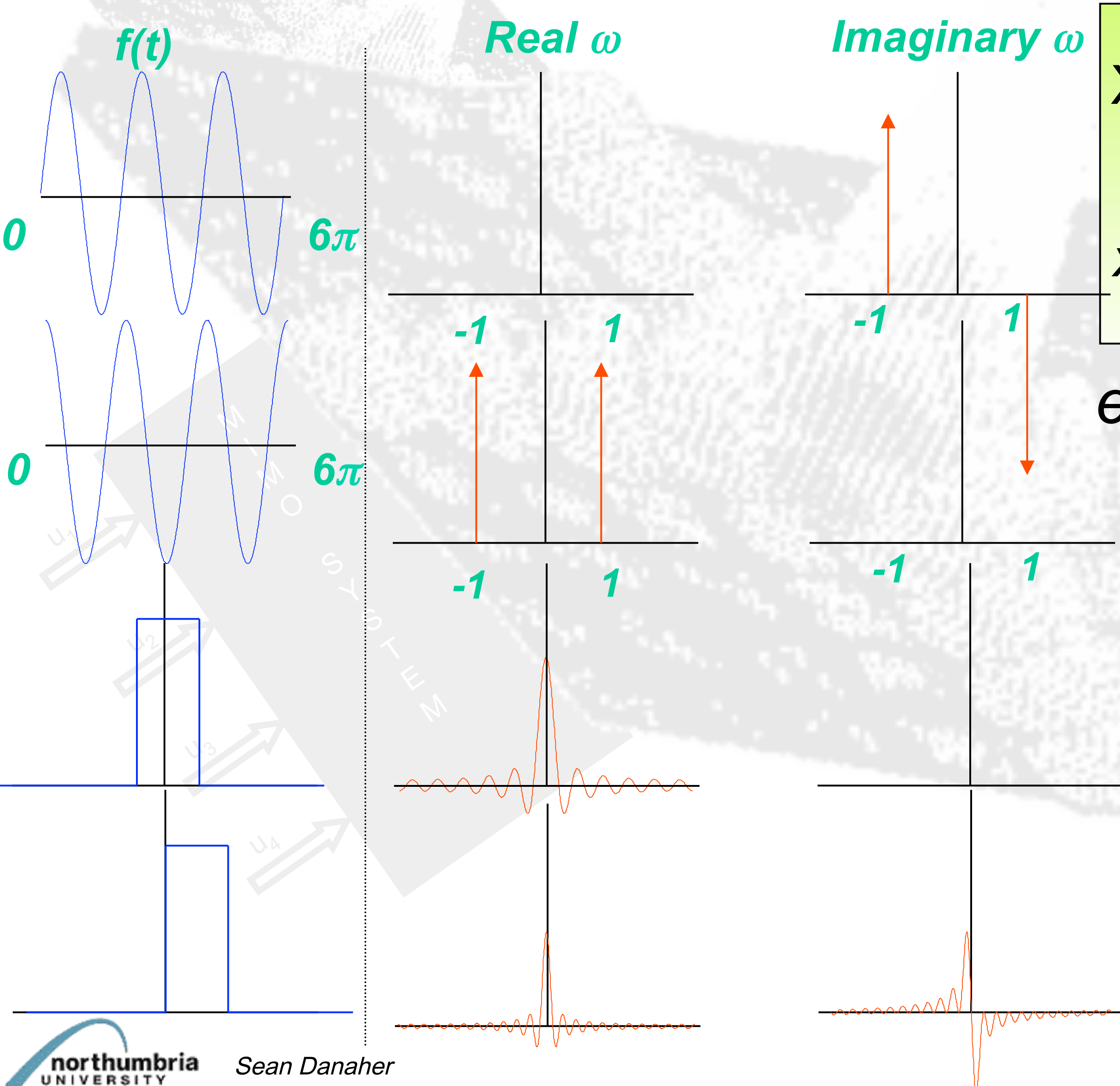
What is Signal Processing?

- Used to model signals and systems where there is correlation between past and current inputs/outputs (in space or time)
- Two broad categories: Continuous and Sampled processes
- A host of techniques Fourier, Laplace, State space, Z-Transform, SVD, wavelets……
- Fast, accurate, robust and easy to implement

Why? What can Signal Processing Do for you?

- *Speed up computation e.g. Acoustic integrals by <u>many</u> orders of magnitude*
- *Design, Understand , simulate SISO systems e.g. Analogue Filters*
- *Design, Understand, simulate MIMO systems e.g. Hydrophones, Microphones, Arianne 5 Rockets etc.*
- *Design, Understand, simulate Digital filters*
- *Design Optimal Filters e.g. Matched Filters*
- *Parametric and non parametric System/Spectral identification/analysis*
- *Design Classification Algorithms*

The Fourier Transform



$$X(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

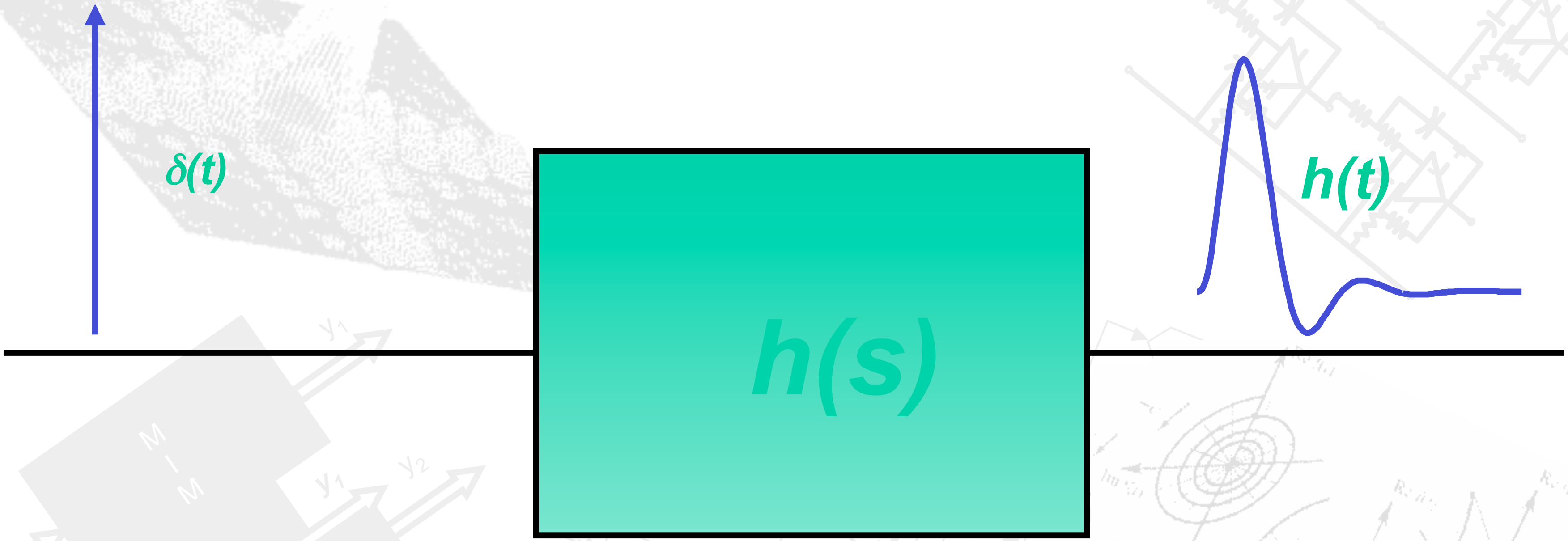
$$x(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega$$

$$e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$$

(.avi)

Sean Danaher

The Impulse Response *h(t)*

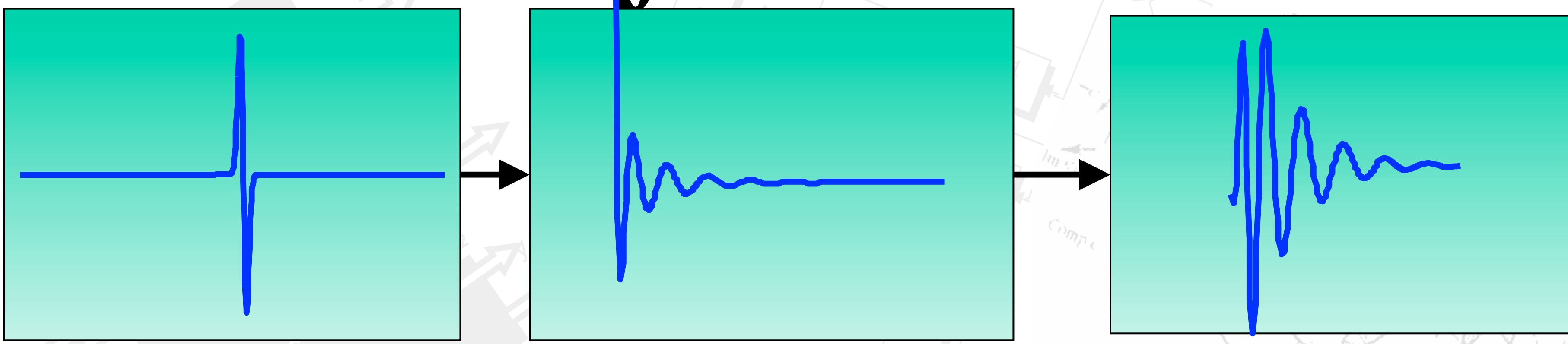


- ***Signals and systems are interchangeable***
- ***A system with an impulse response h(s) is equivalent to a signal h(s).***

The Convolution Integral

- Given a signal *s*(*t*) and a system with an impulse response *h*(*t*) then *y(t)* is given by

$$y(t) = \int_{-\infty}^{\infty} s(t)h(t-\tau)d\tau$$



s(t) Bipolar Acoustic Pulse

h(t) Impulse response high pass filter (SAUND)

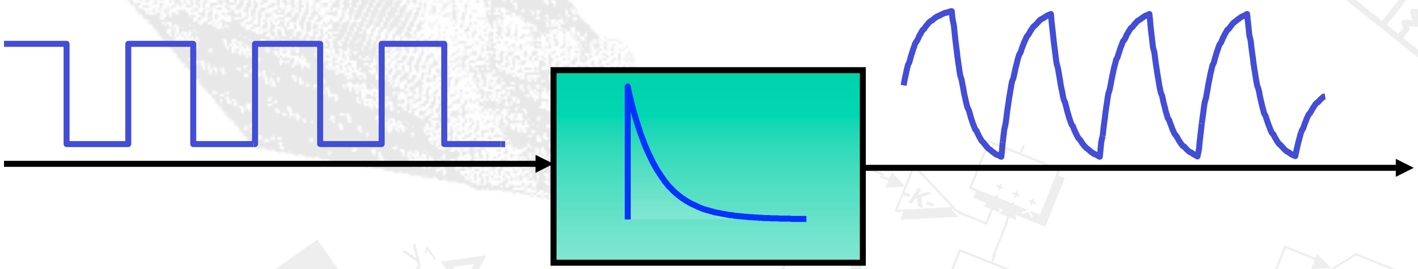
y(t) Electrical Pulse

The nasty impulse response is caused by a rapidly varying phase response: Different frequencies pass through the filter at different speeds

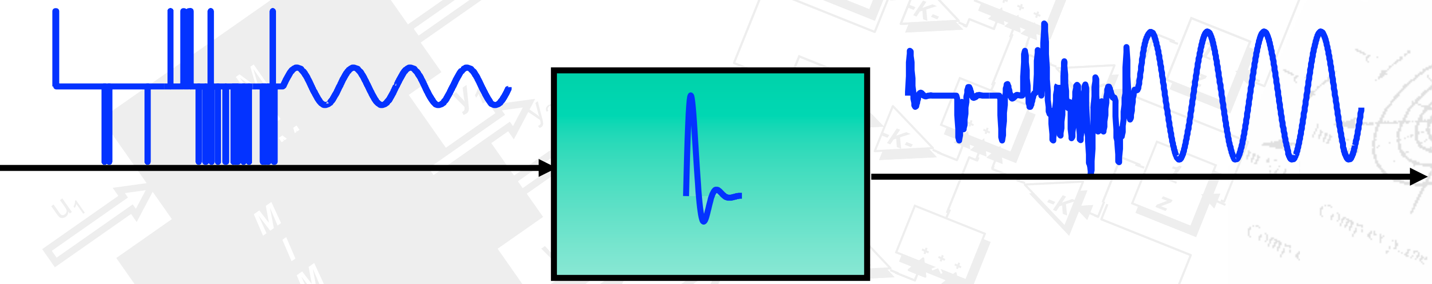
Acoustic Integrals in Water and Ice

Prof. Sean Danaher
University of Northumbria,
Newcastle UK

A Few examples



RC circuit



Mixed signal

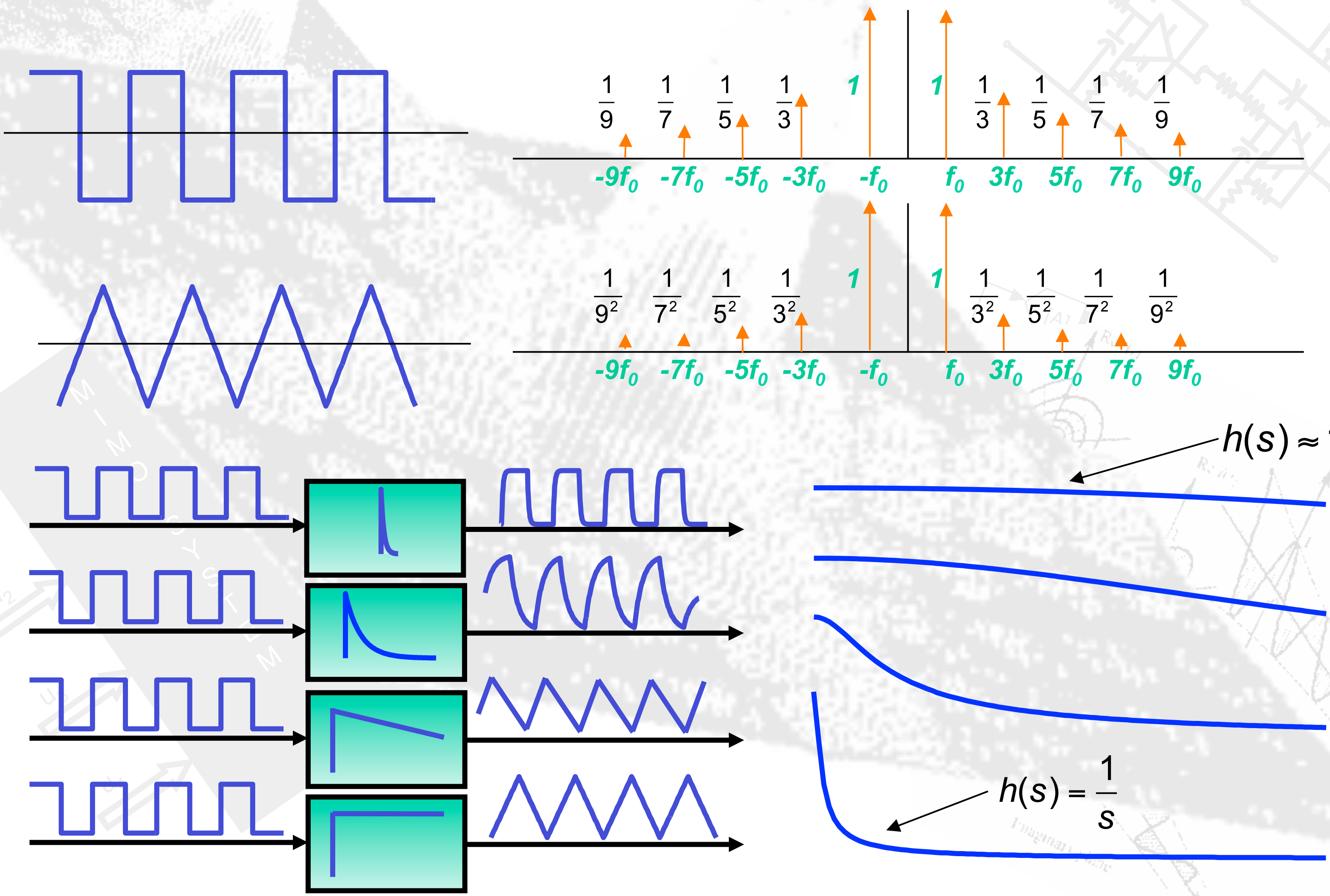
2d

Convolution Theorem

Convolution in the time domain is Multiplication in the frequency domain (and visa versa)

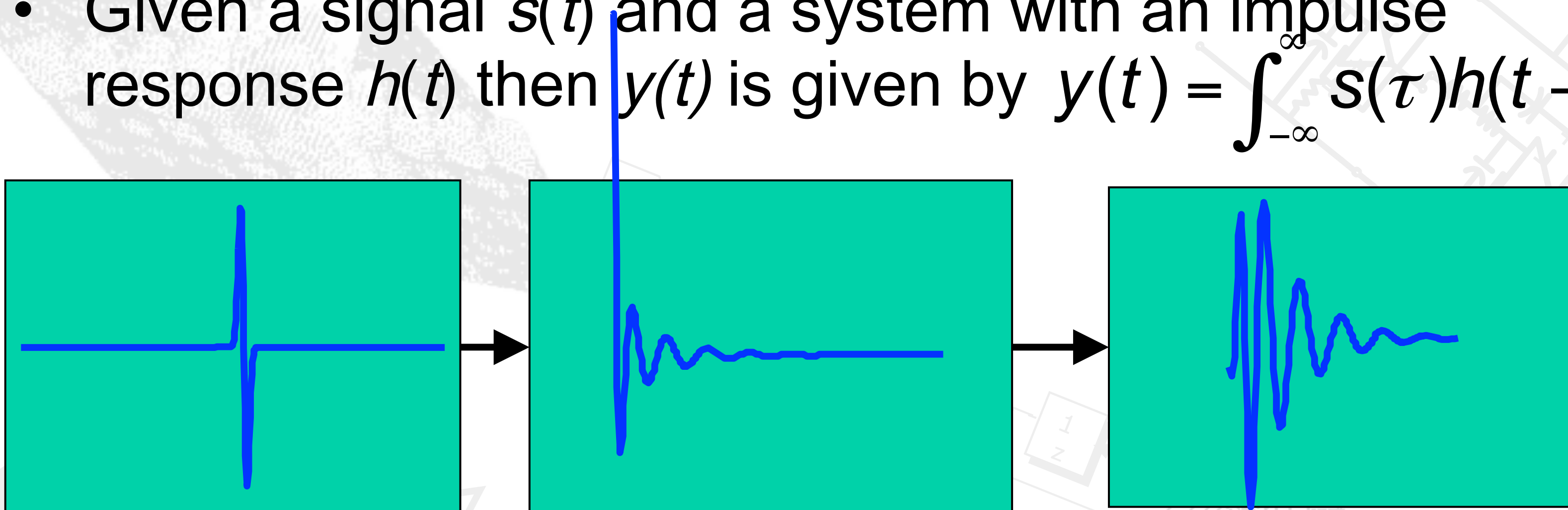
- Conceptually very useful
- Convolution very expensive computationally ($o=n^2$)
  - Convert the signal and impulse response to the frequency domain (Fourier Transform)
  - Provided the number of points $n=2^m$ very efficient FFTs $o=n \log n$
  - Multiply and take Inverse Transform

Convolution Theorem Example



Calculation of output

- Given a signal $s(t)$ and a system with an impulse response $h(t)$ then $y(t)$ is given by $y(t) = \int_{-\infty}^{\infty} s(\tau)h(t-\tau)d\tau$



$s(t)$ Bipolar Acoustic Pulse

$h(t)$ Impulse response of Hydrophone

$y(t)$ Electrical Pulse

Alternatively take the Fourier transform of the signal (both amplitude and phase information)
Multiply the amplitude information by the frequency response of the Hydrophone
Adjust the phase delays
Take inverse Fourier transform

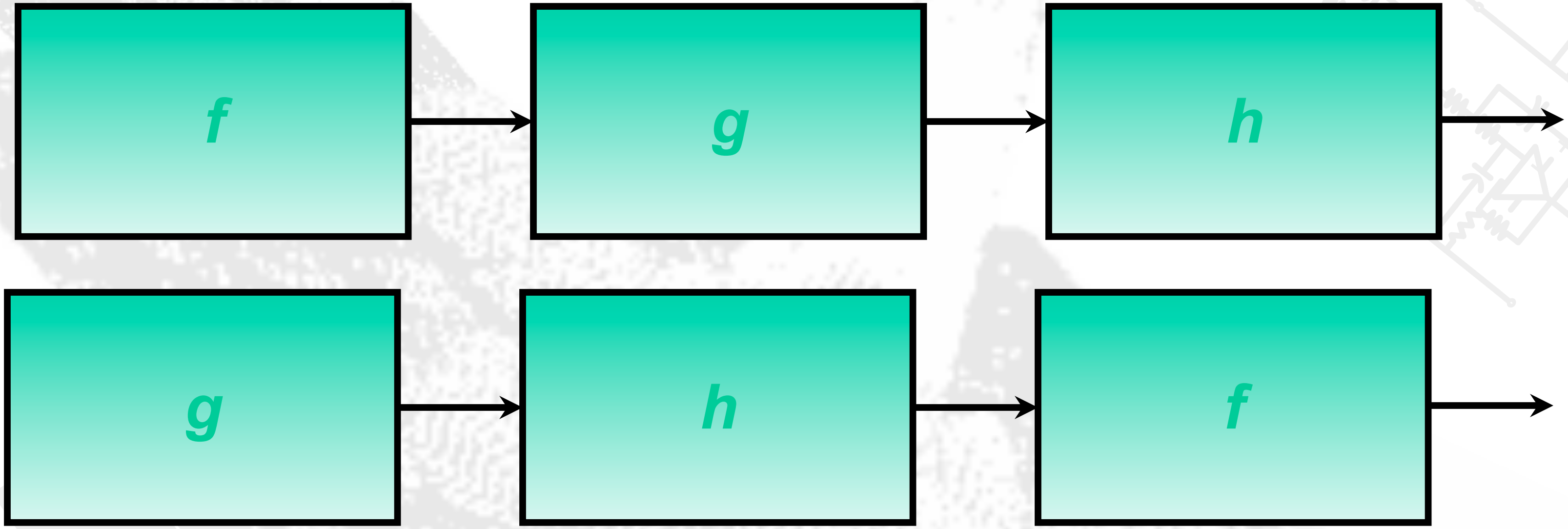
The Frequency Response

- If $h(s)$ is known, the frequency response can be determined simply by putting in $s=i\,\omega$
- Alternatively a Fourier Transform can be used directly on the signal $h(t)$

The frequency response is complex: it contains phase information.
"The importance of the Phase response can not be overstated!" – Paraphrased from SAUND paper (June 2004)

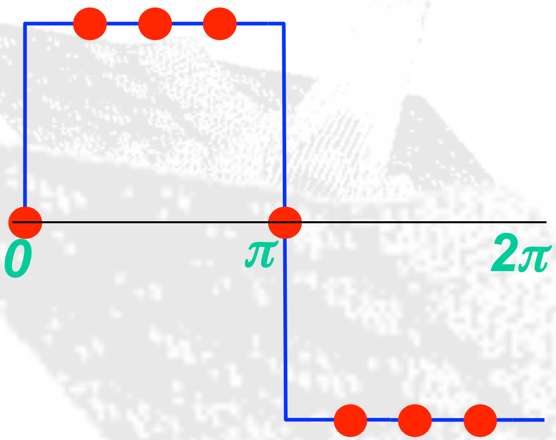
$$y = \frac{du}{dt} \rightarrow h(s) = s \qquad y = \int u\,dt \rightarrow h(s) = \frac{1}{s}$$

Order Not important



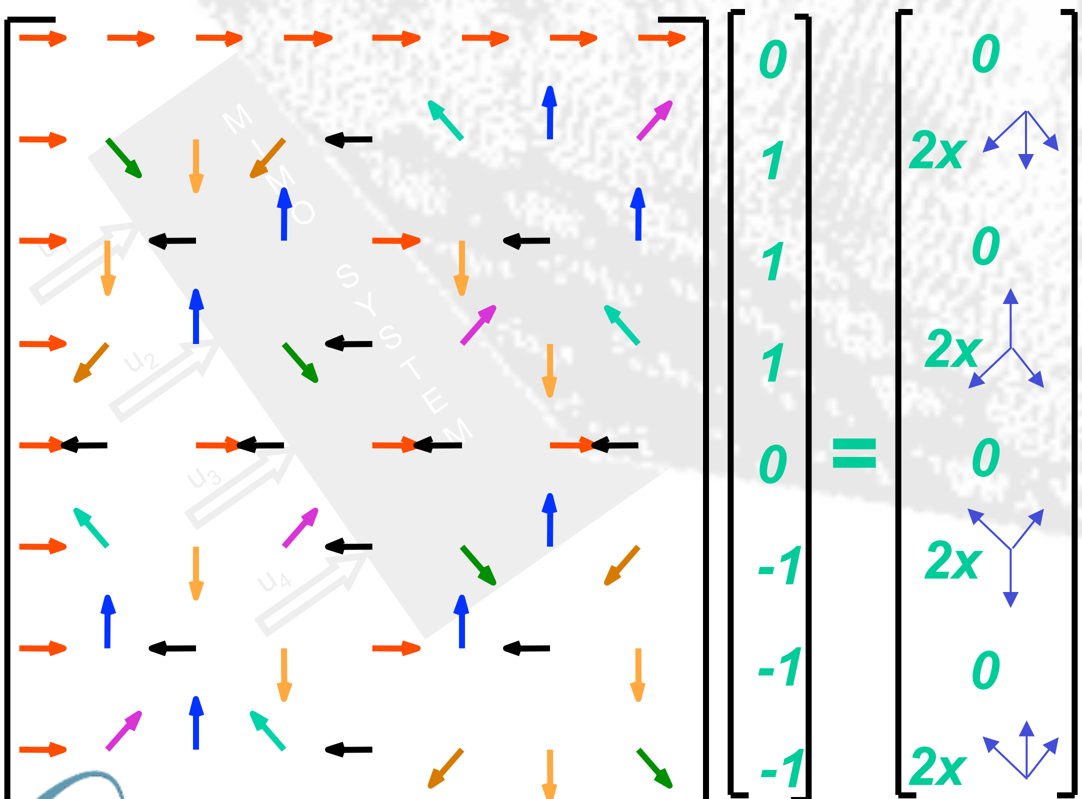
For linear systems /processes order irrelevant to result but often not to Computational speed!

DFT Example and FFT



Sample n points. Eight for this example but 1024 more typical

Multiply by DFT Matrix



Cooley and Tukey introduced the first FFT algorithm in 1965

Uses redundancy in the multiplies using the "Butterfly"

Get O (n log n)

Rather than O(n²)

Latest algorithm FFTW – fastest Fourier transform in the West

Sean Danaher

Autocorrelation

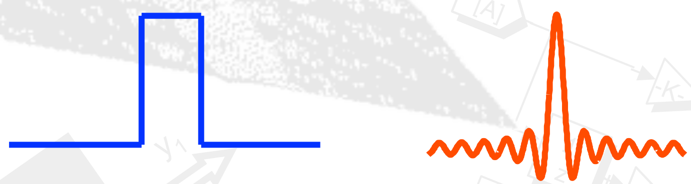
$$R_{xx}(t) = \int_{-\infty}^{\infty} x(t)x(t+\tau)d\tau$$

When τ =0 proportional to the total energy in the signal

Looks very like convolution. Indeed identical apart from the sign.
Use a filter whose impulse response is a time reversed copy of the original signal

Wiener Khinchin Theorem

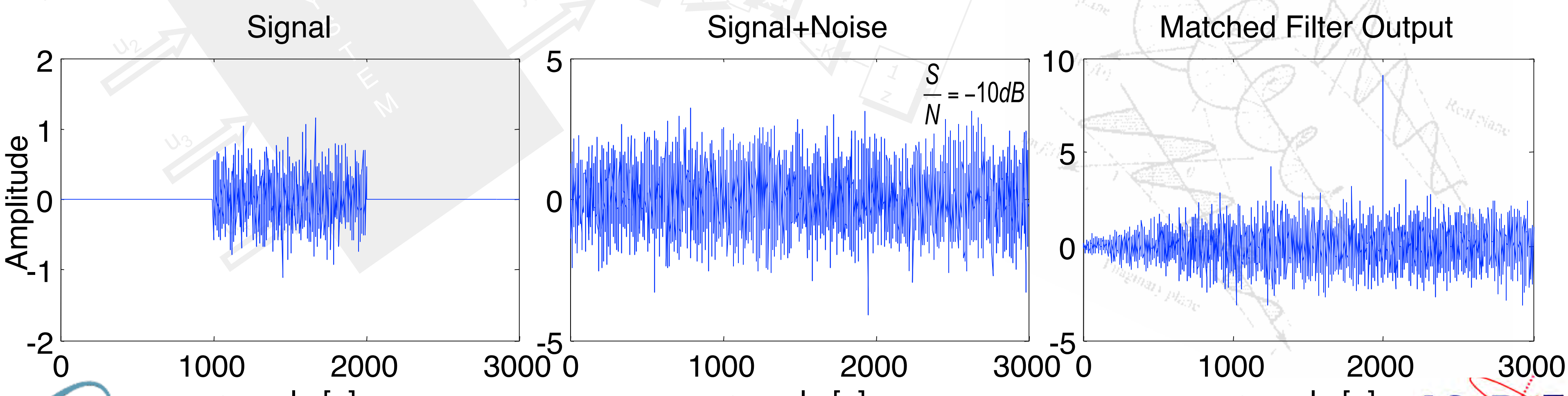
The Autocorrelation function and PSD are Fourier transform pairs



Chirp ->Top Hat in Frequency domain
->sin(x)/x autocorrelation
Resolution depends on frequency sweep not pulse duration

For best resolution we need a signal with an autocorrelation of $\delta(t)$.

- $\delta(t)$ obviously
- Also White noise

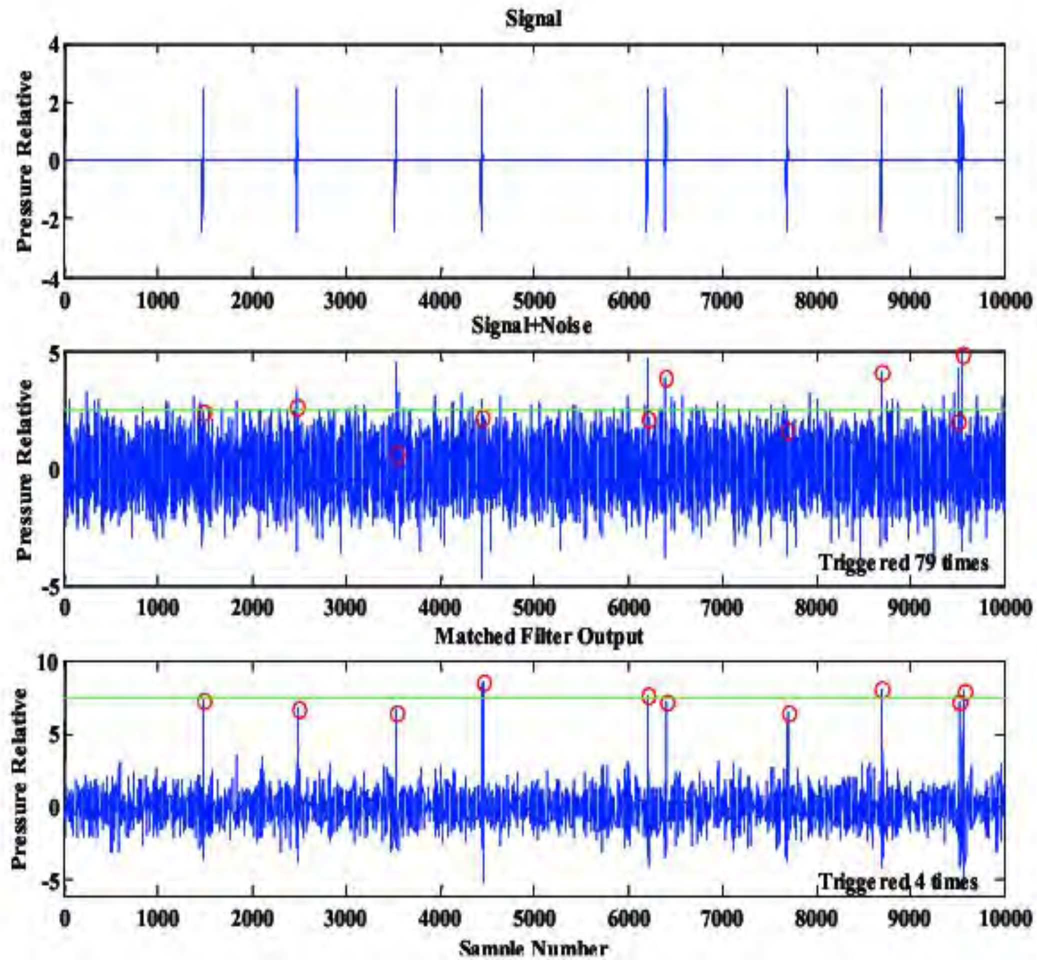


northumbria UNIVERSITY
Sean Danaher
ACoRVE

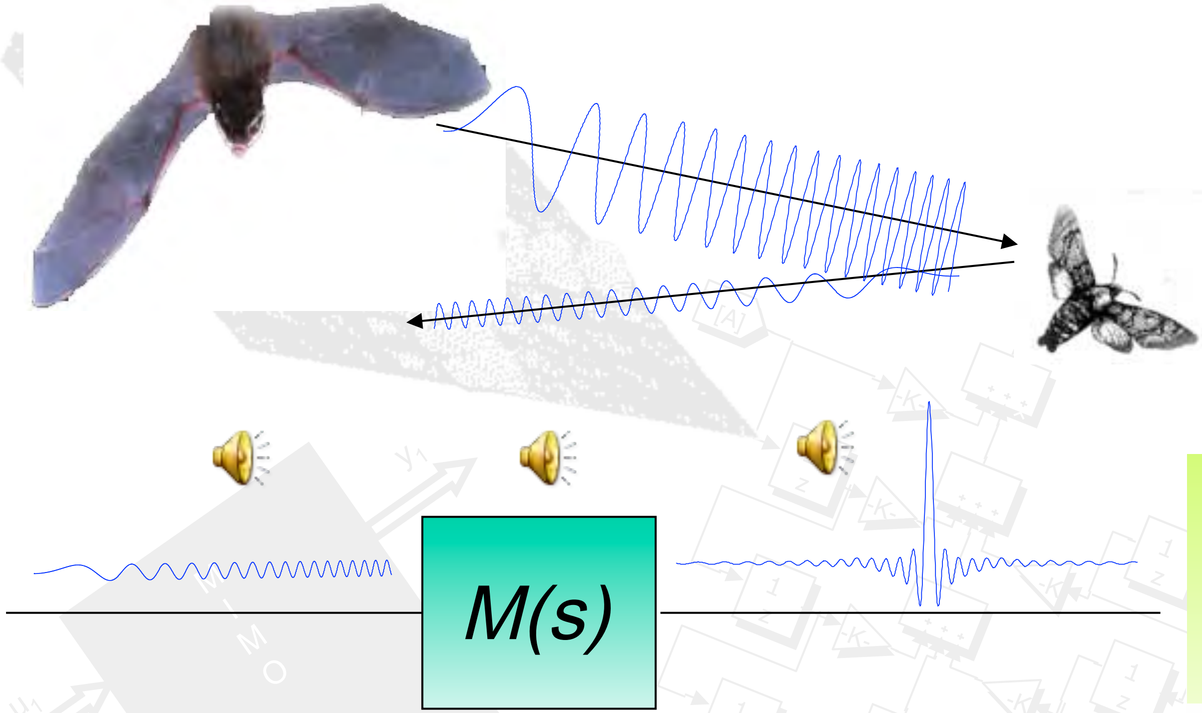
Matched Filter

The inverse of an all pole filter is guaranteed to be stable as it is an all zero filter (does not use feedback) This is a pre-whitening filter

- Provided the Noise is white a matched filter has an impulse response which is that of the time reversed signal.
- If the noise is not white run both the signal and noise through a pre-whitening filter
- Design a filter who's impulse response is the time reversed filtered signal



Matched Filters



$$E \propto \frac{1}{r^4}$$

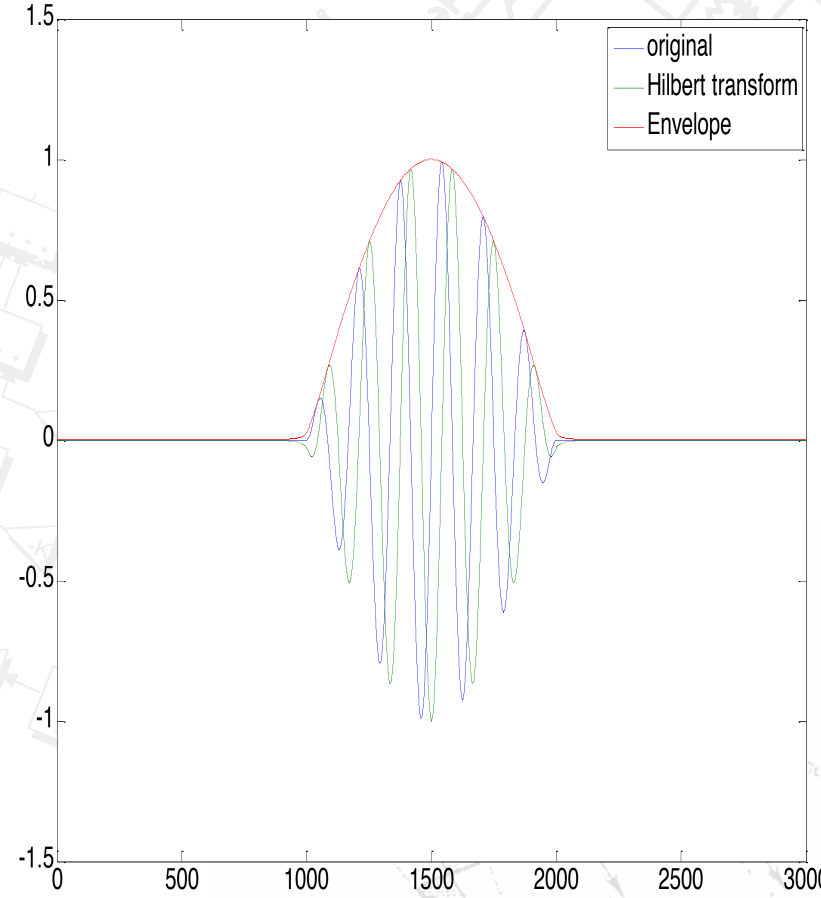
$$Accuracy \propto \frac{1}{\tau}$$

Need a very high peak power or Signal Processing

The frequency response matches that of the signal
The phase response is adjusted to slow the transit of the frequencies which arrive first so the signal "bunches up"

The Hilbert Transform

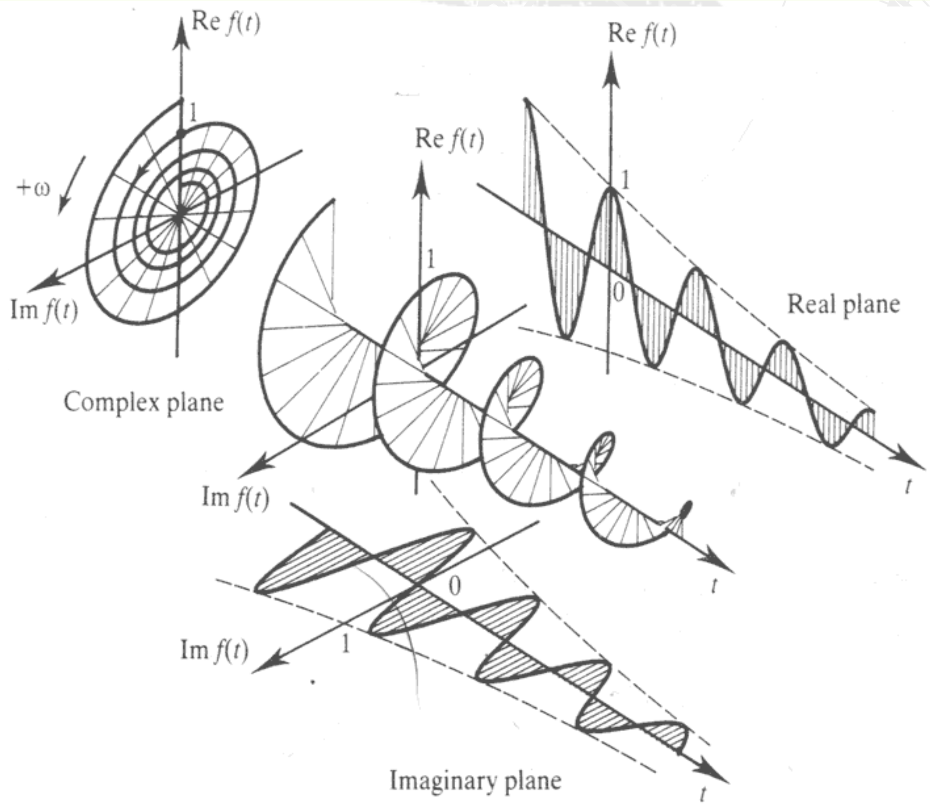
- *Converts Sine to Cosine*
- *Useful for Envelope detection*
- *Use a FFT to convert to frequency domain*
- *Multiply by j (i)*
- *IFFT*

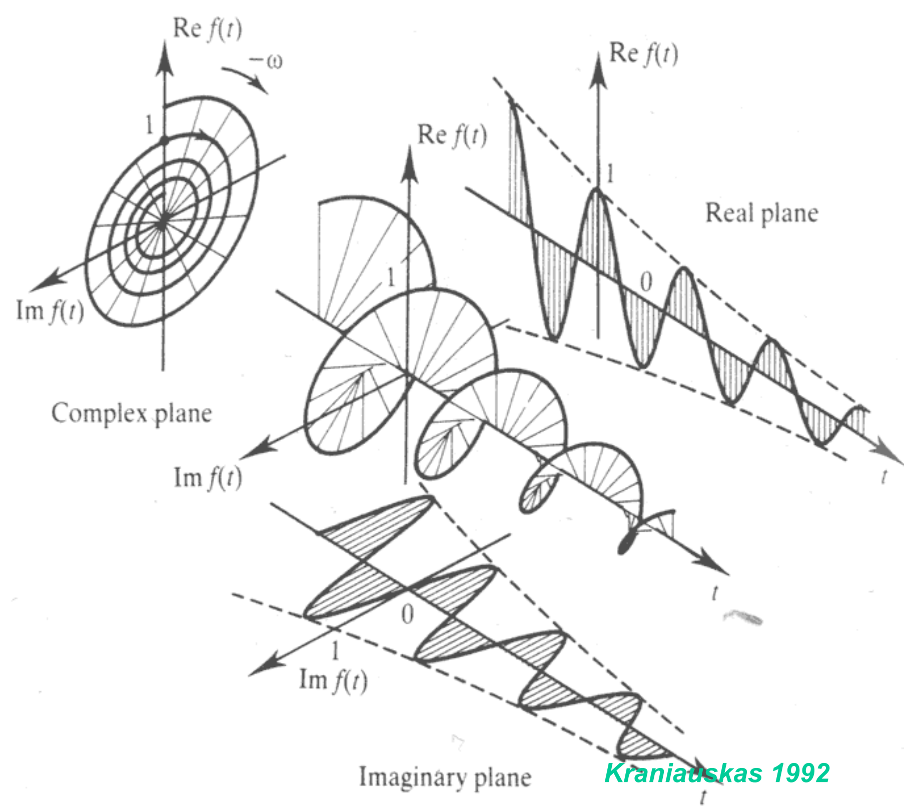


Laplace

The Complex Signal

The signal e^{st} can be generalised to $s=\sigma+i\omega$ This includes decaying and growing exponentials etc.

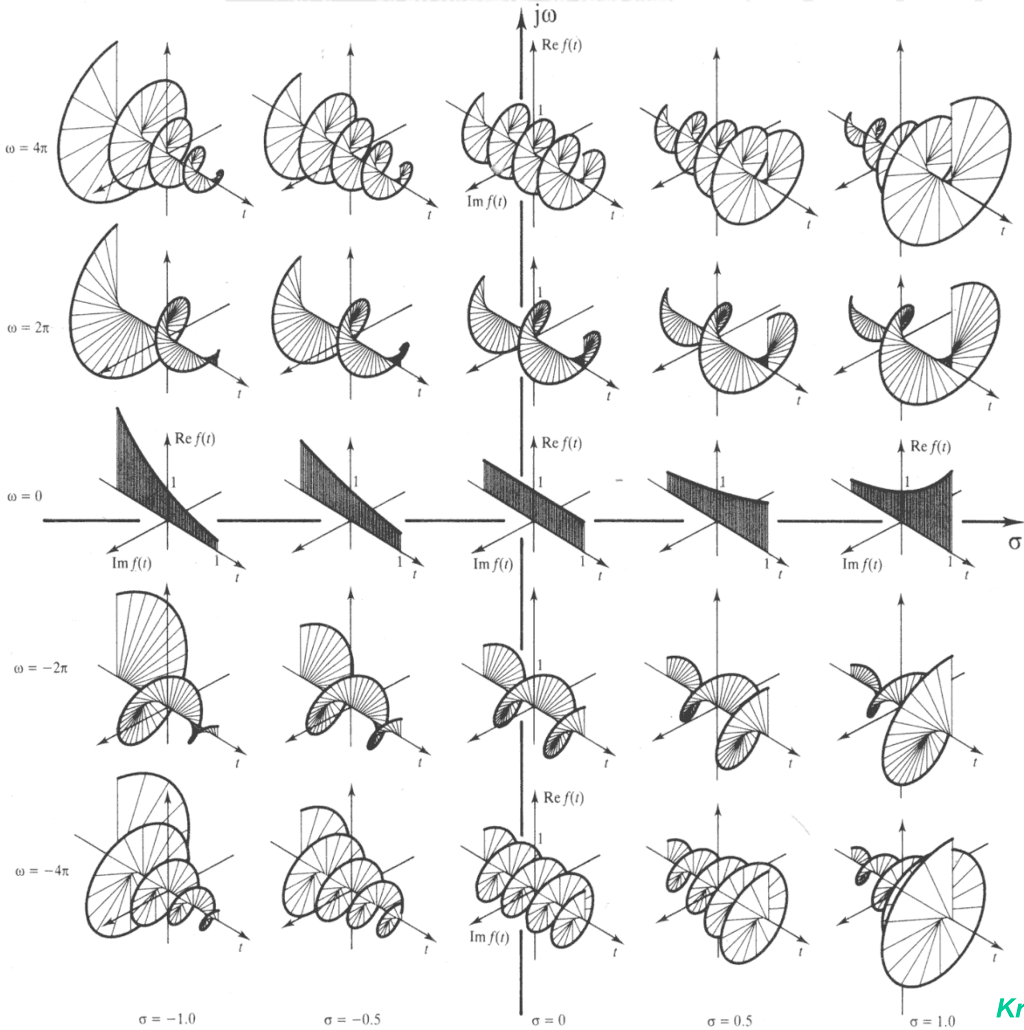




Kraniauskas 1992

Continuous Systems

$$a_n \frac{d^n y}{dt^n} + \ldots + a_2 \frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m u}{dt^m} + \ldots + b_2 \frac{d^2 u}{dt^2} + b_1 \frac{du}{dt} + b_0 u$$

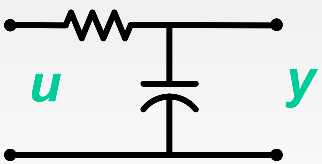


$$h(s) = \frac{b_m s^m + \ldots + b_2 s^2 + b_1 s + b_0}{a_n s^n + \ldots + a_2 s^2 + a_1 s + a_0}$$

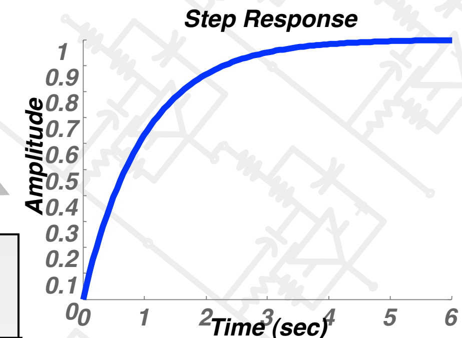
If we can represent the system in terms of a differential equation then we can write h(s) straight down Graphically we represent this by a complex surface on the s-plane

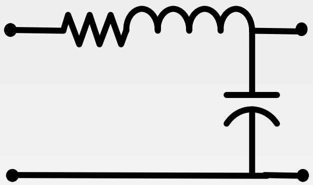
Kraniauskas 1992

Some simple Transfer Functions

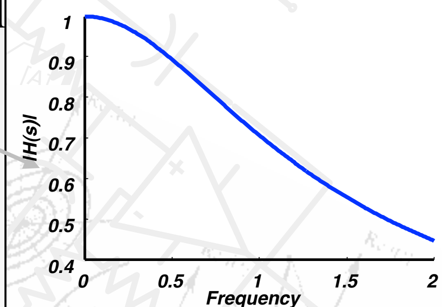


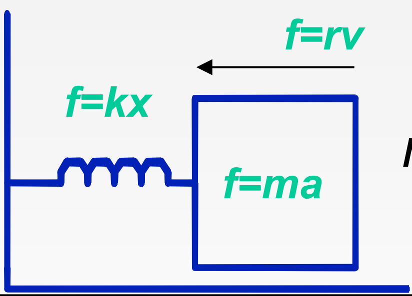
$$RC\frac{dy}{dt} + y = u \qquad h(s) = \frac{1}{RCs + 1}$$



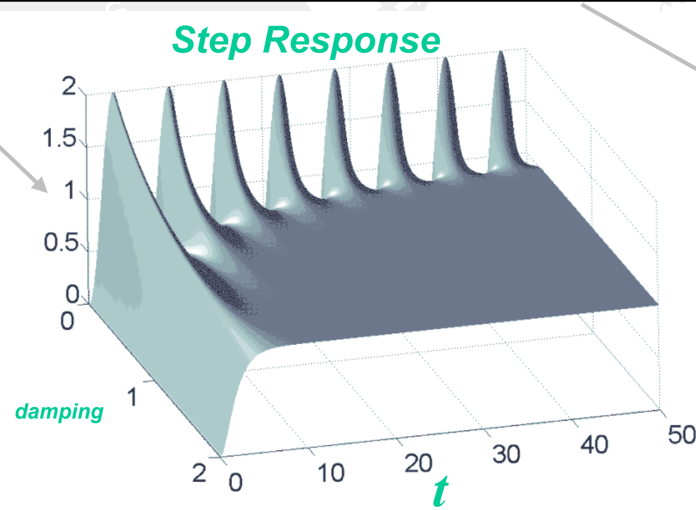


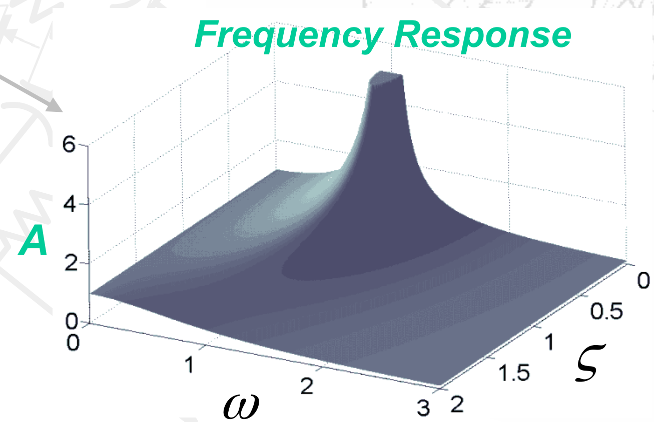
$$LC\frac{d^2y}{dt^2} + RC\frac{dy}{dt} + y = u \quad h(s) = \frac{1}{LCs^2 + RCs + 1}$$





$$m\frac{d^2y}{dt^2} + r\frac{dy}{dt} + ky = u \quad h(s) = \frac{1}{ms^2 + rs + k}$$





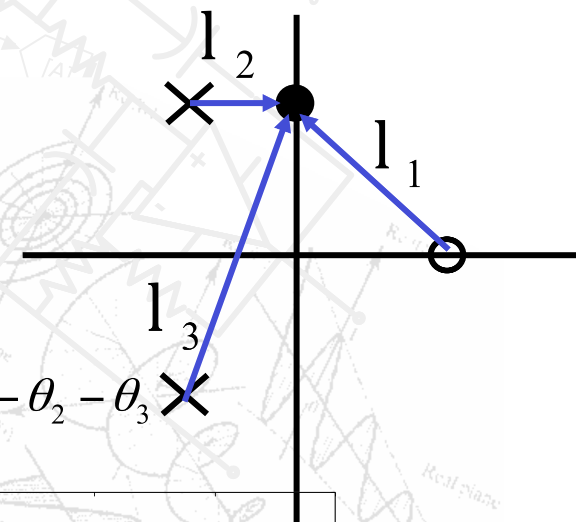
Poles and Zeros

$$h(s)=\frac{b_m s^m+\ldots+b_2 s^2+b_1 s+b_0}{a_n s^n+\ldots+a_2 s^2+a_1 s+a_0}$$

$$=\frac{(s-z_m)(s-z_{m-1})\ldots.(s-z_1)}{(s-p_n)(s-p_{n-1})\ldots..(s-p_1)}$$

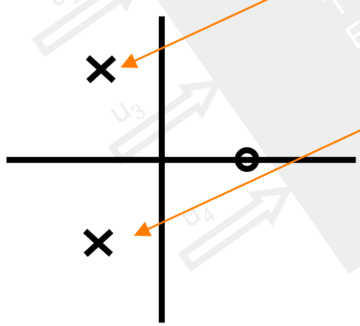
Like all polynomials the transfer function can be factorised
Zeros are roots of the Numerator
Poles are Roots of the Denominator

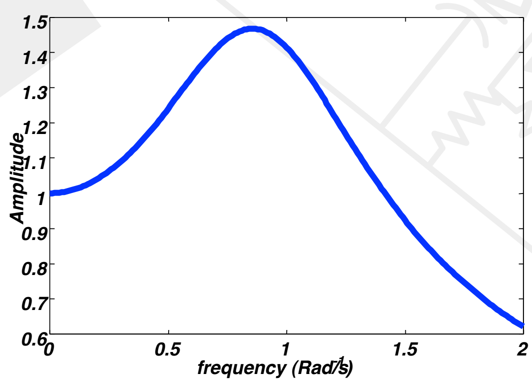
$$=(s-z_m)(s-z_{m-1})\ldots.(s-z_1)\frac{1}{(s-p_n)}\frac{1}{(s-p_{n-1})}\ldots.\frac{1}{(s-p_1)}$$

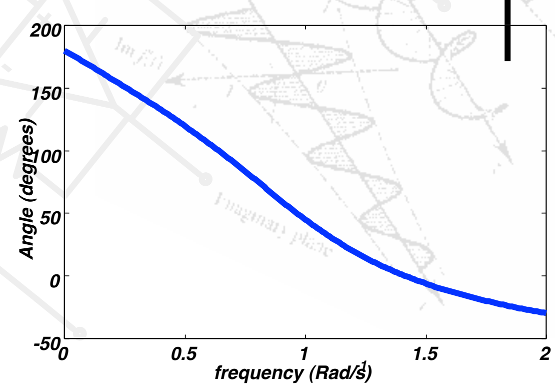


$$h(s)=\frac{s-1}{s^2+s+1}=\frac{s-1}{(s-(-0.5+0.866i))(s-(-0.5-0.866i))}$$

$$|h(s)|=\frac{l_1}{l_2 l_3},\angle h(s)=\theta_1-\theta_2-\theta_3$$







Filter Design

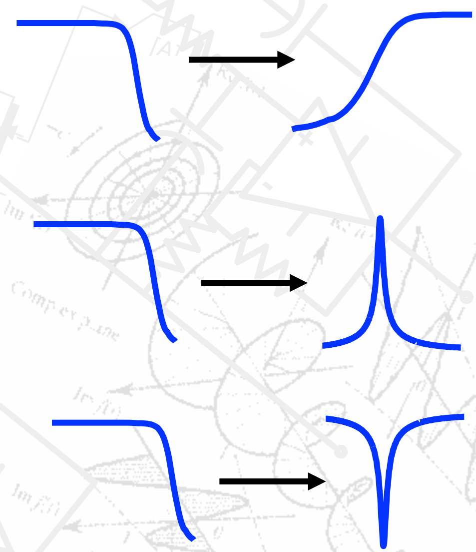
Design Low pass Filter with cut off frequency of ω=1. Transform to type:

Frequency Scaling: $s \rightarrow \dfrac{s}{\omega_0}$

Low pass to high pass: $s \rightarrow \dfrac{\omega_0}{s}$

Low pass to band pass $s \rightarrow \dfrac{\omega_0^{\,2}}{Bws}\left(\left(\dfrac{s}{\omega_0}\right)^2 + 1\right)$

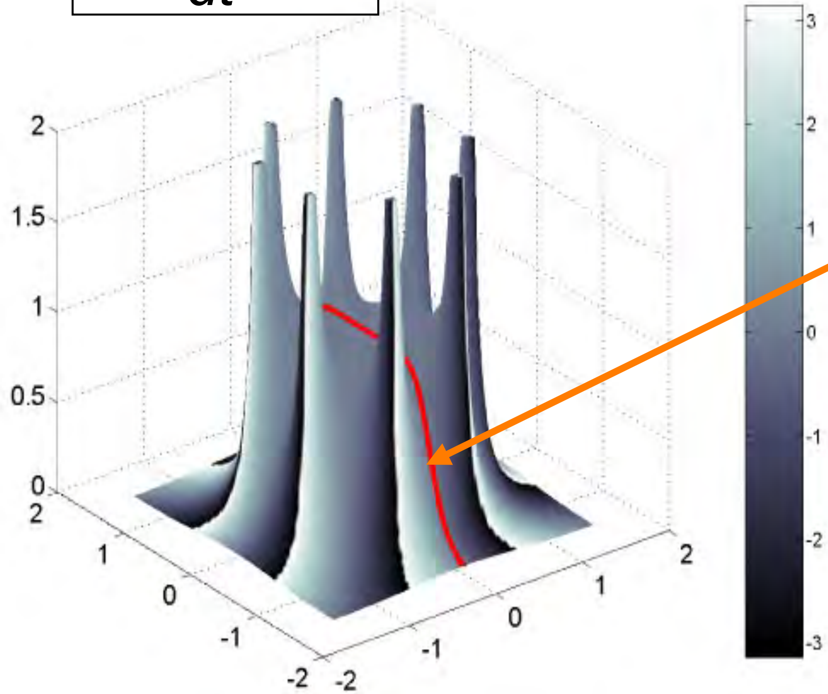
Low pass to band stop $s \rightarrow \dfrac{Bws}{\omega_0^{\,2}}\left(\left(\dfrac{s}{\omega_0}\right)^2 + 1\right)^{-1}$

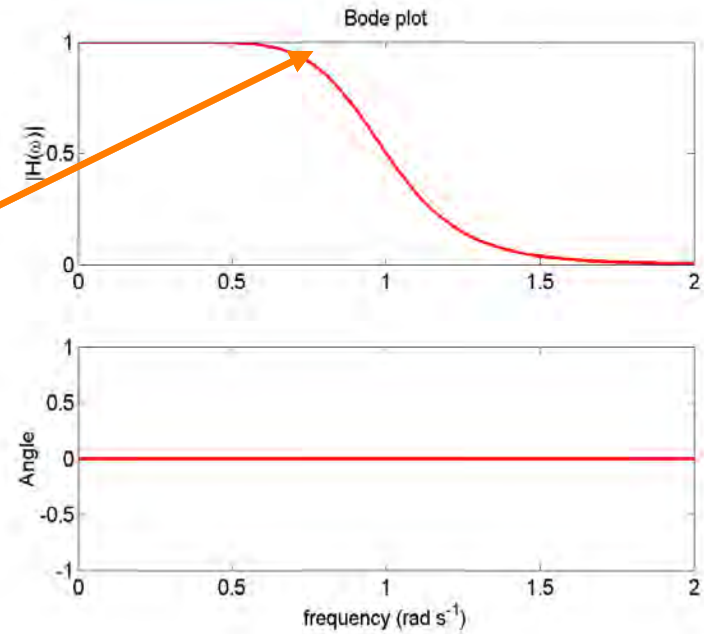


$$y+\frac{d^8 y}{dt^2}=u$$

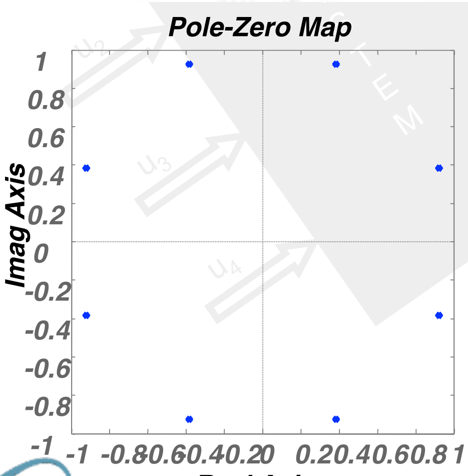
A perfect filter?

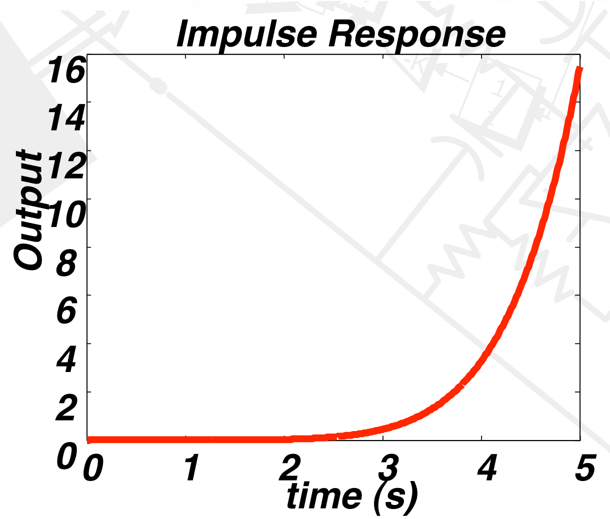
$$h(s)=\frac{1}{s^8+1}$$





Frequency response determine d by running along the iω axis





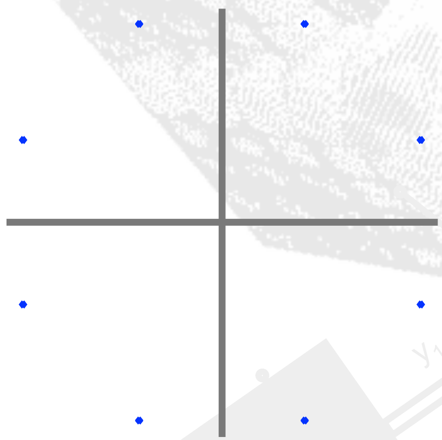
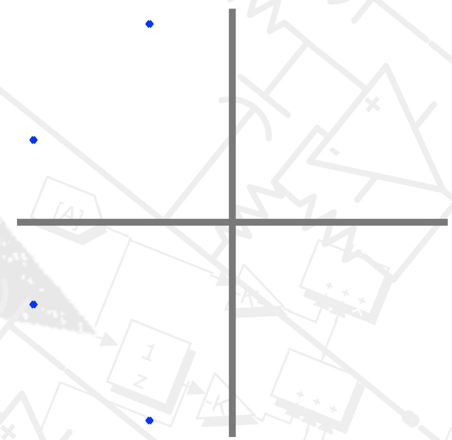
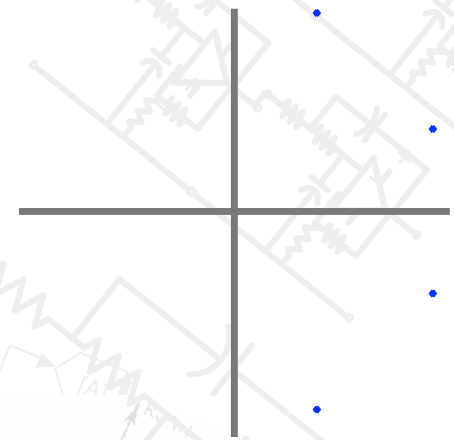
Perfect Phase Response But unstable due to poles on LHS of S-Plane

northumbria UNIVERSITY

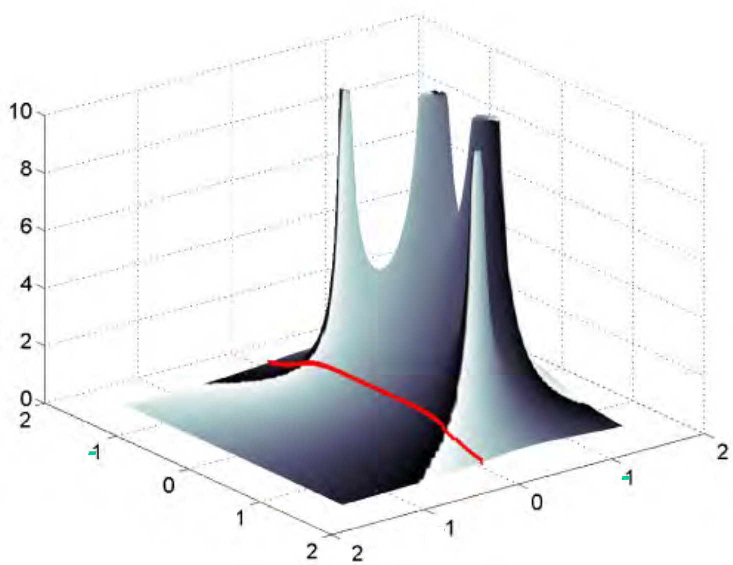
Sean Danaher

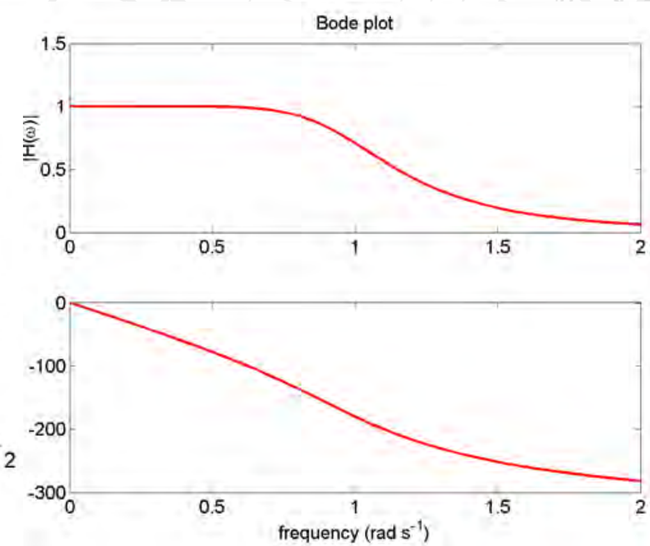
ACoRVE

Stability Comes at a Cost

 =  × 

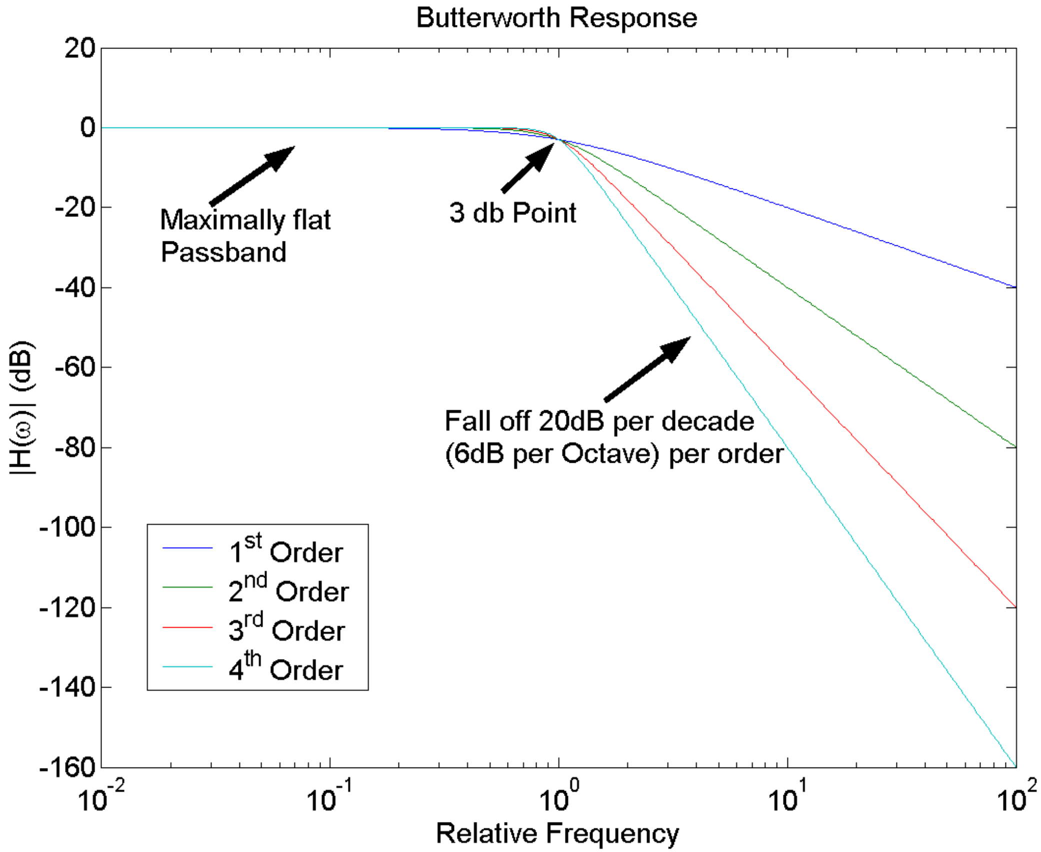
$$\boxed{\frac{1}{1+s^8}} = \frac{1}{s^4+2.613s^3+3.414s^2+2.613s+1} \times \frac{1}{s^4-2.613s^3+3.414s^2-2.613s+1}$$





System now stable but a rapidly varying phase response BUTTERWORTH

Butterworth Filter

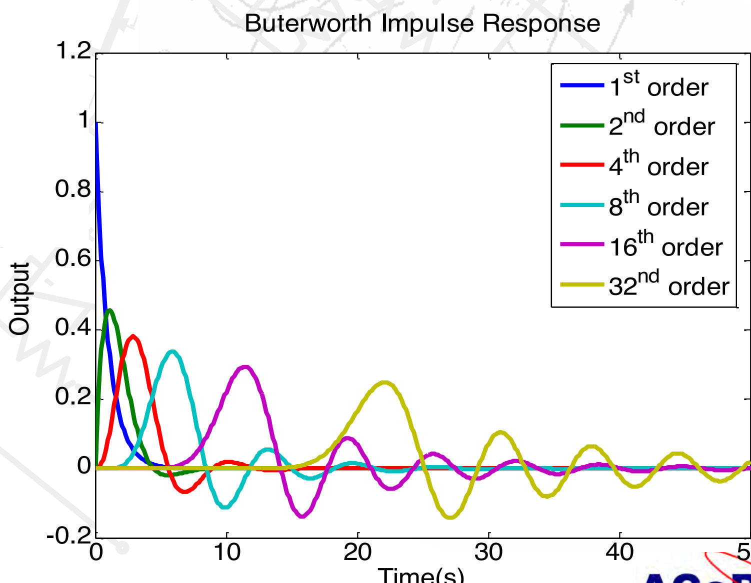


1930s

$$|H(\omega)| = \frac{1}{\sqrt{1+\omega^{2n}}}$$

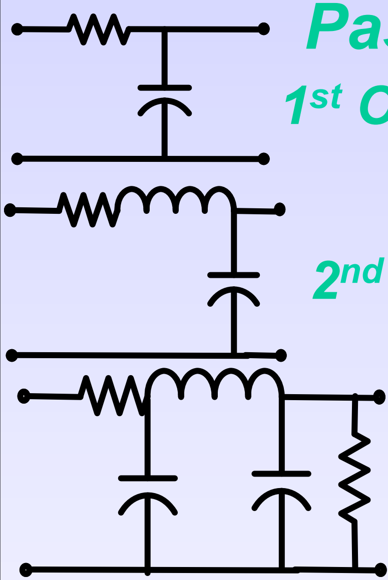
Stable
Good Frequency Response

Note Impulse response
- ***Output Delay***
- ***Oscillation***



Implementation

At acoustic frequencies easiest to use op-amps. Expensive to make high precision lossless inductors



Passive Filters

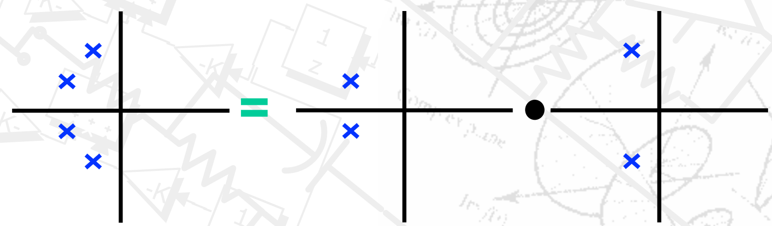
1st Order Familiar RC

2nd Order LRC

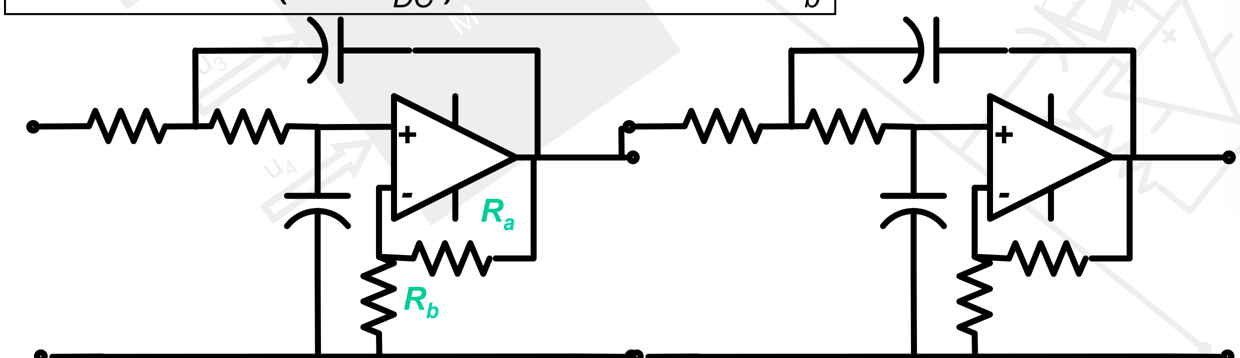
3rd Order "π"

$$\frac{1}{s^4+2.6131s^3+3.4142s^2+2.6131s+1}$$

$$=\frac{1}{s^2+0.7654s+1}\bullet\frac{1}{s^2+1.8478s+1}$$



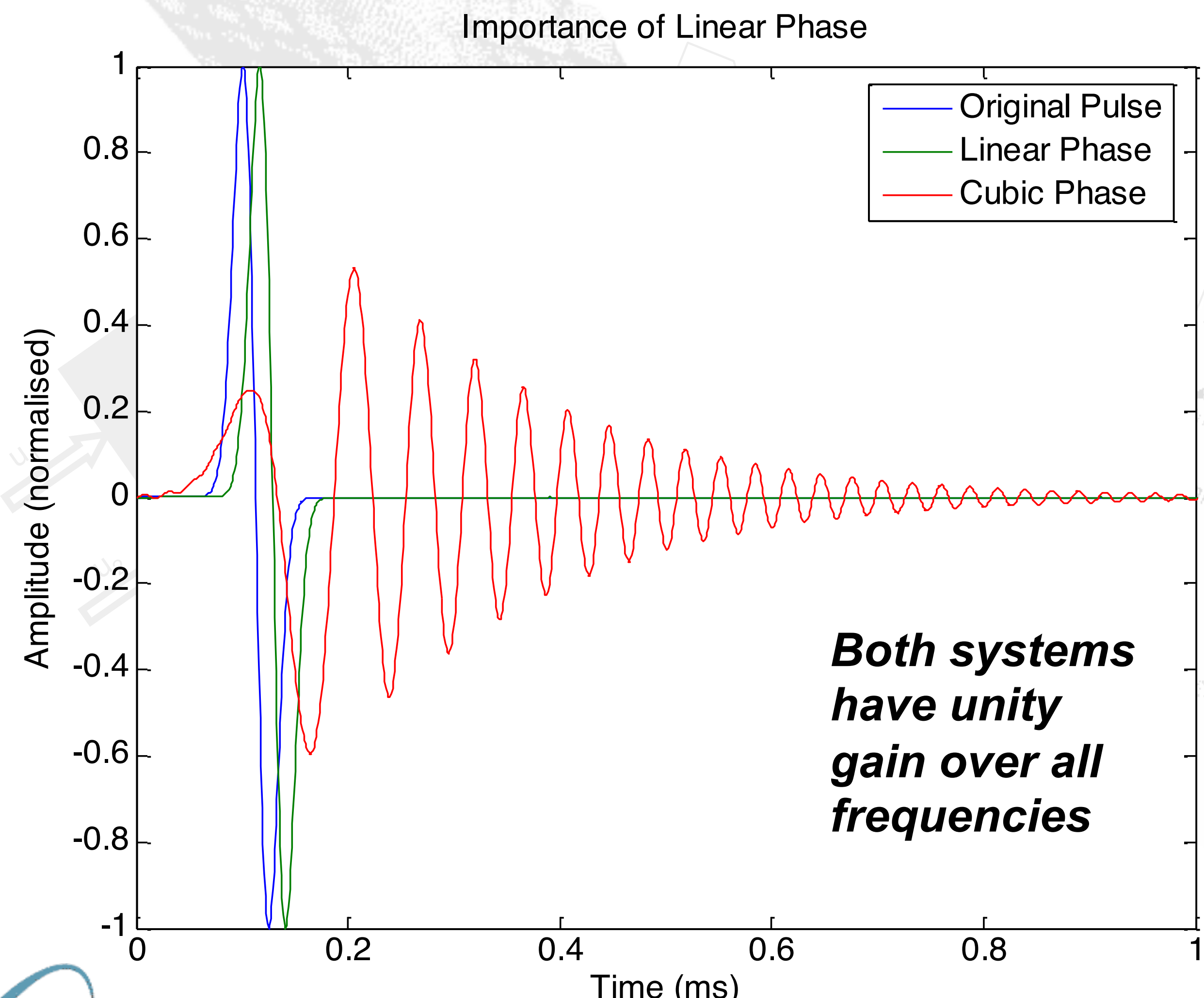
$$|h(s)|=\frac{A_{DC}}{s^2+(3-A_{DC})s+1},\quad A_{DC}=1+\frac{R_a}{R_b}$$



4th Order Butterworth active with 2 second order sections

The importance of linear Phase

Will a bipolar acoustic pulse give a bipolar electrical pulse?



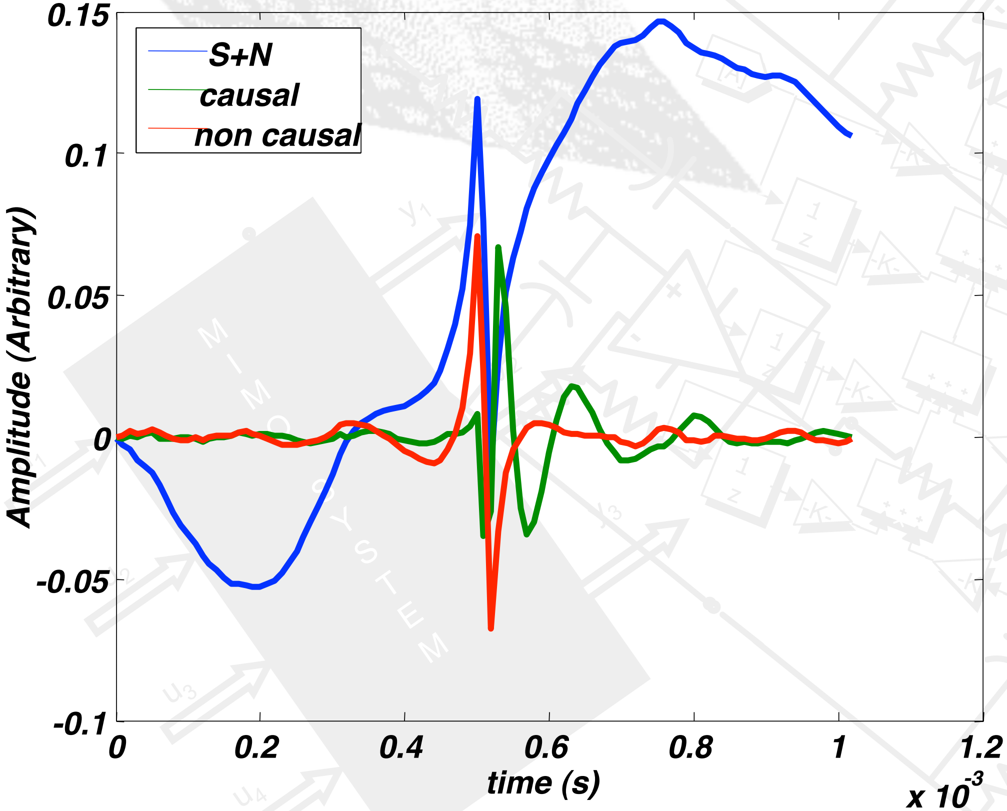
Provided the phase response of the hydrophone/amplifier/ filter system is linear over the region of interest then the pulse is simply delayed. If the phase response is non linear then distortion will occur.

A linear phase response gives a constant group delay this is analogous to group velocity in continuous systems

$$\frac{d\theta}{d\omega} \equiv \frac{d\omega}{dk}$$

Recovering Phase information

e.g. RonaData

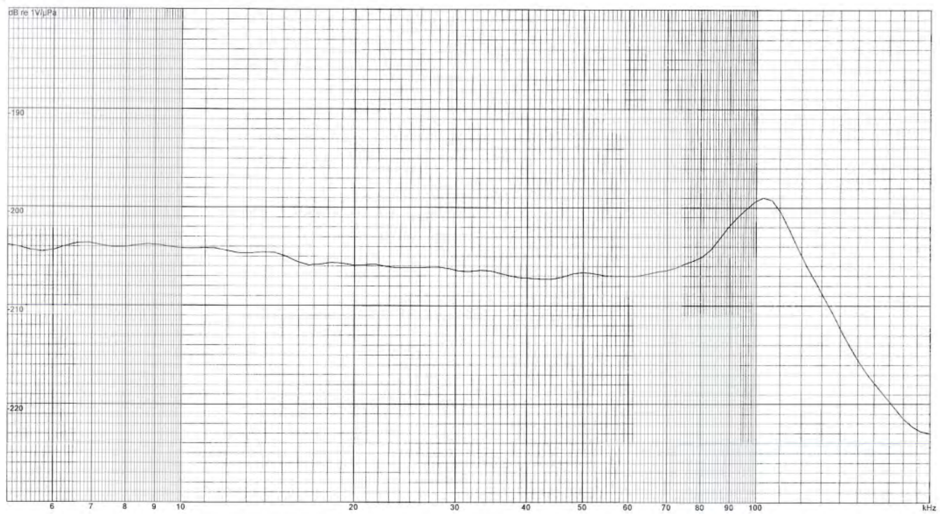


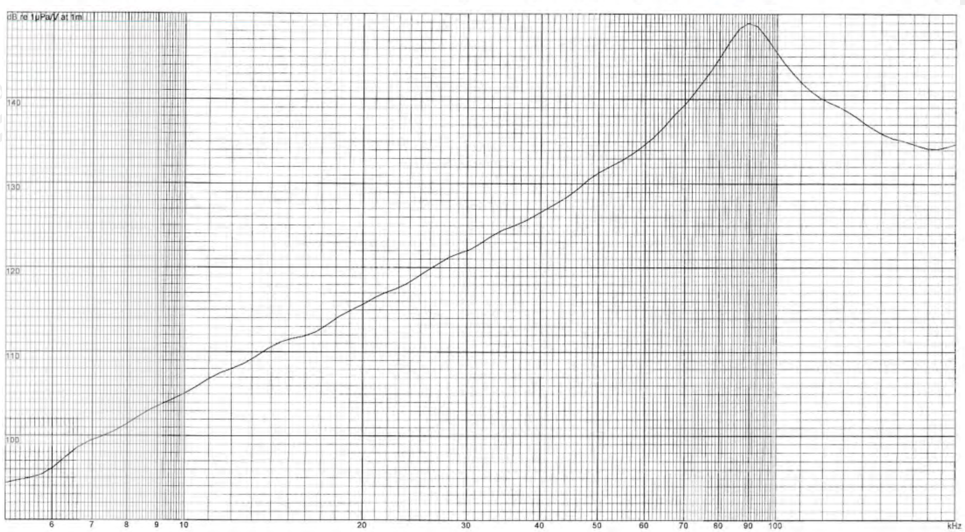
It is trivial to design digital filters which have a constant group delay $d\phi/d\omega$ and hence no phase distortion

If however we know the filter response e.g. Butterworth we can run the data through the filter backwards.

This increases the order of the filter by a factor of 2

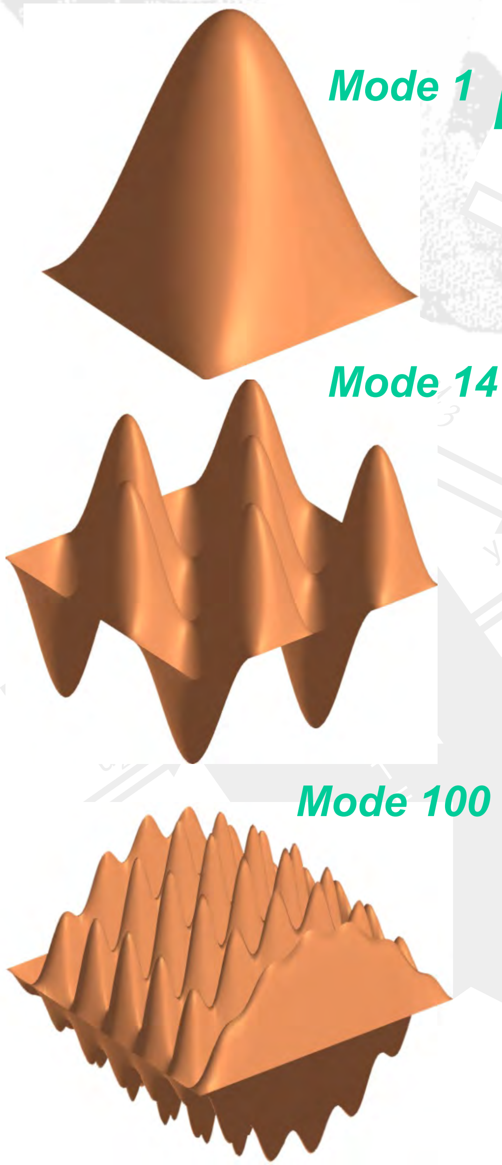
transmit more difficult





Fairly flat response when in receive mode, but in transmit the amplitude typically depend on hydrophone acceleration:
Velocity drives acoustic production
Impedance matching between hydrophone and water typically gives another derivative term

State Space Analysis



MIMO systems

Mixed Mechanical/electrical models etc

Matrix Based Method

$$\dot{X} = AX + BU$$

$$Y = CX + DU$$

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ . \\ . \\ \dot{x}_n \end{pmatrix} = A \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{pmatrix} + B \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ . \\ . \\ u_k \end{pmatrix}$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ . \\ . \\ y_m \end{pmatrix} = C \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ . \\ . \\ x_n \end{pmatrix} + D \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ . \\ . \\ u_k \end{pmatrix}$$

All modern control algorithms use SS methods.

Pictured 20000 state simulation of a square membrane: (100x100 masses 2 states x and v) A Matrix 400x10^{6} elements

SS Implementation

	States	Inputs
States	*A*	*B*
Outputs	*C*	*D*

A is of size States × States
B is of size States × Inputs
C is of size Outputs × States
D is of size Outputs × Inputs

States are degrees of freedom of the system. Things that store energy

- ***Capacitor Voltages***
- ***Inductor currents***
- ***Positions and Velocities of masses***

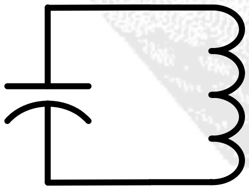
$$\text{if} \quad x = V_c \Rightarrow I_C = C\frac{dV}{dt} = C\dot{x}$$

$$\text{if} \quad x = I_L \Rightarrow V_L = L\frac{di}{dt} = L\dot{x}$$

$$\text{if} \quad x = \text{position} \Rightarrow \dot{x} = \text{velocity}$$

$$\text{if} \quad x = \text{velocity} \Rightarrow \dot{x} = \text{acceleration}$$

Simple Example LC



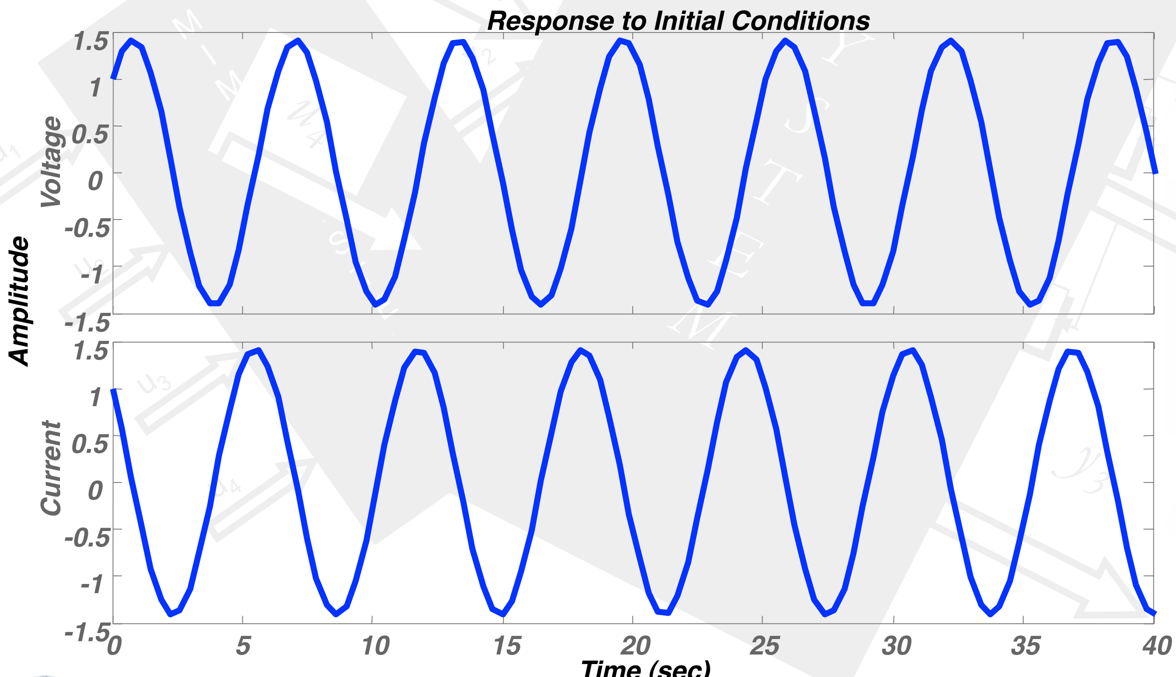
$$I_c = C\frac{dV_c}{dt} \qquad V_L = (-)L\frac{dI_L}{dt}$$

$$V_L = V_c \qquad I_L = -I_C$$

$$x_1 = V_C \qquad C\dot{x}_1 = x_2 \qquad y_1 = x_1$$

$$x_2 = I_L \qquad L\dot{x}_2 = -x_1 \qquad y_2 = x_2$$

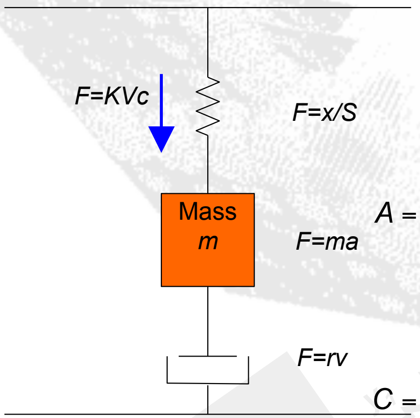
$$\mathbf{A} = \begin{pmatrix} 0 & \frac{1}{C} \\ -\frac{1}{L} & 0 \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



Maths simple to implement Example shows the behaviour with an initial 1V on the Capacitor (1F) and 1A flowing through the inductor (1H)

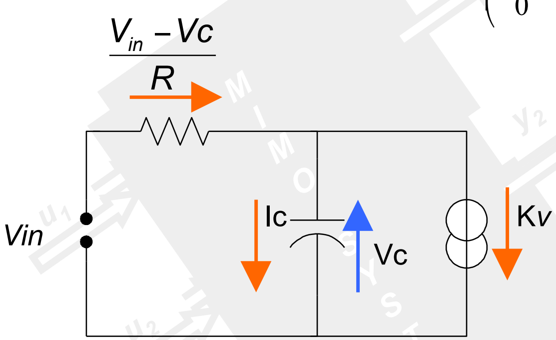
Simple Hydrophone Model

$$x_1 = V_C,\quad x_2 = v,\quad x_3 = x$$

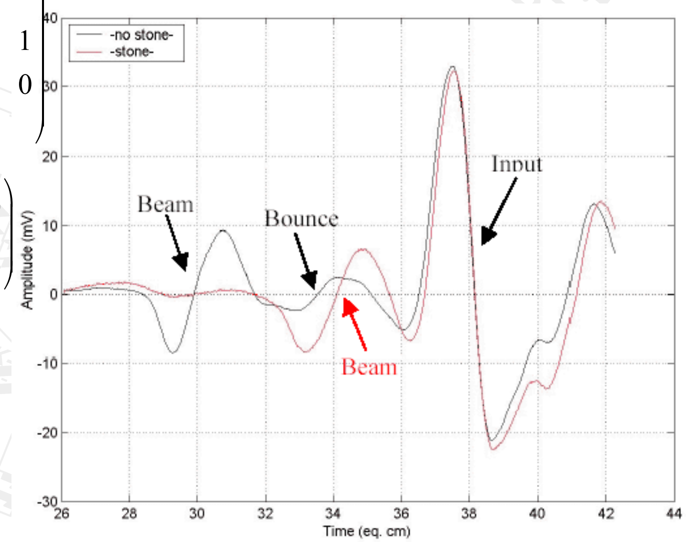


$$A = \begin{pmatrix} -\dfrac{1}{RC} & -\dfrac{K}{C} & 0 \\ \dfrac{K}{m} & -\dfrac{r}{m} & -\dfrac{1}{Sm} \\ 0 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} \dfrac{1}{RC} & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

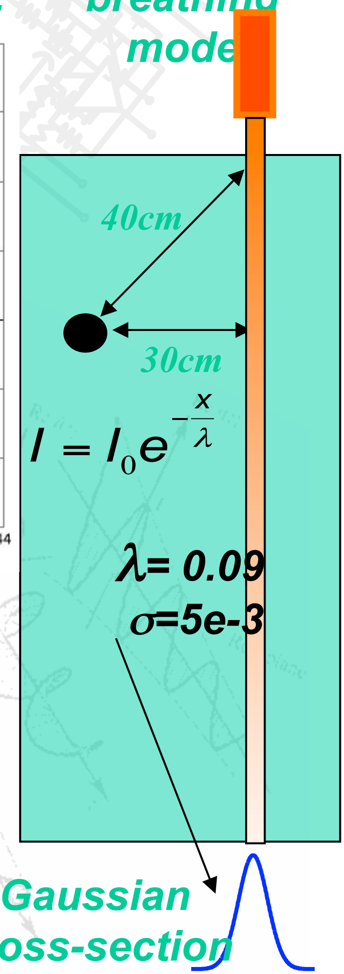
$$C = \begin{pmatrix} -\dfrac{1}{R} & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} \dfrac{1}{R} & 0 \\ 0 & 0 \end{pmatrix}$$



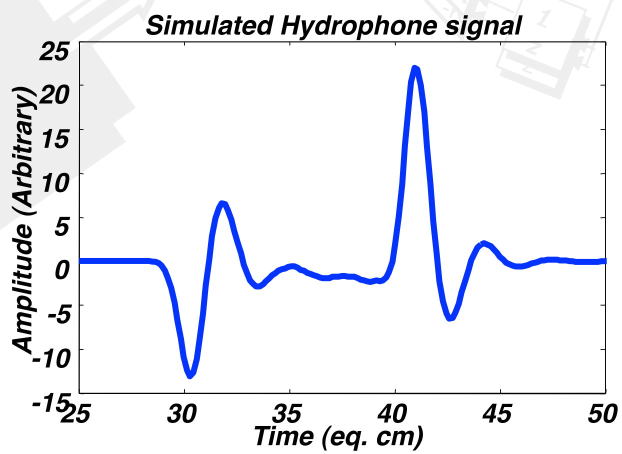
Heart of Hydrophone
Piezo electric crystal
Piezo electric effect



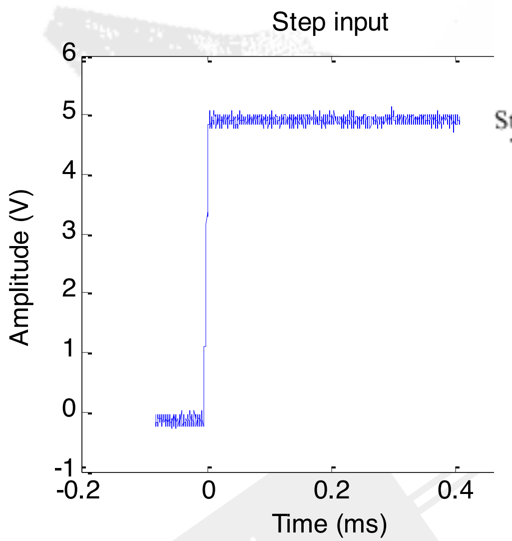
Omni works in breathing mode

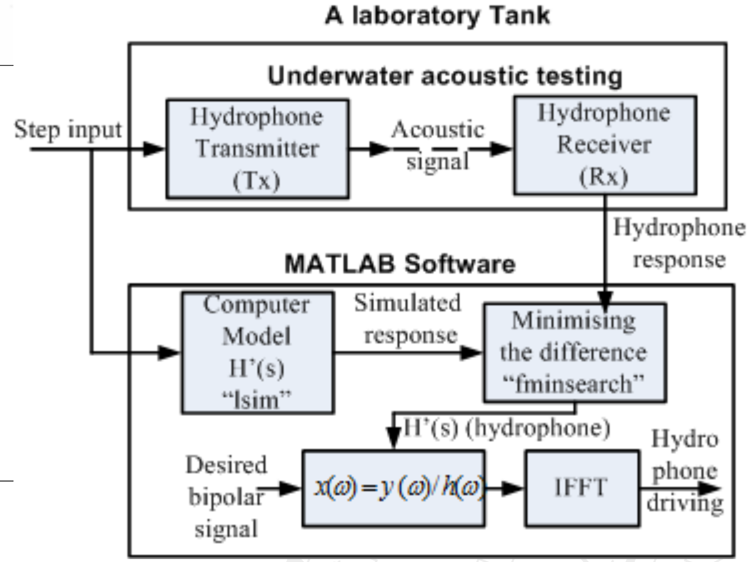


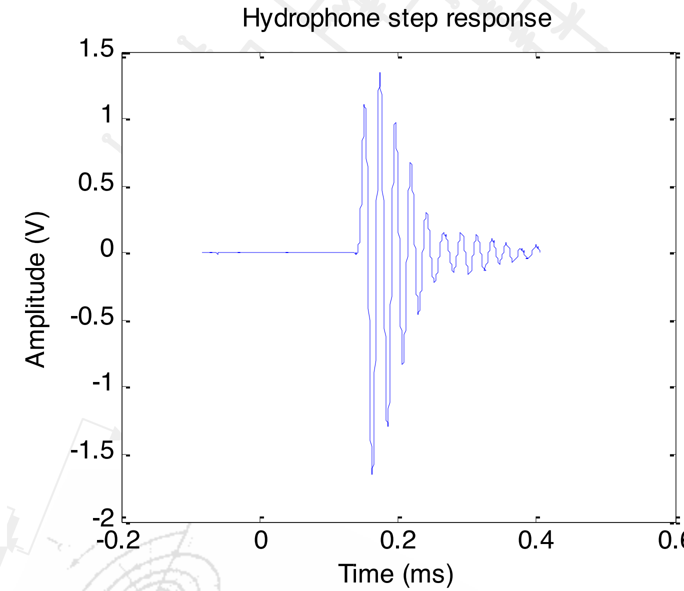
3rd order simulation. Do we need higher?



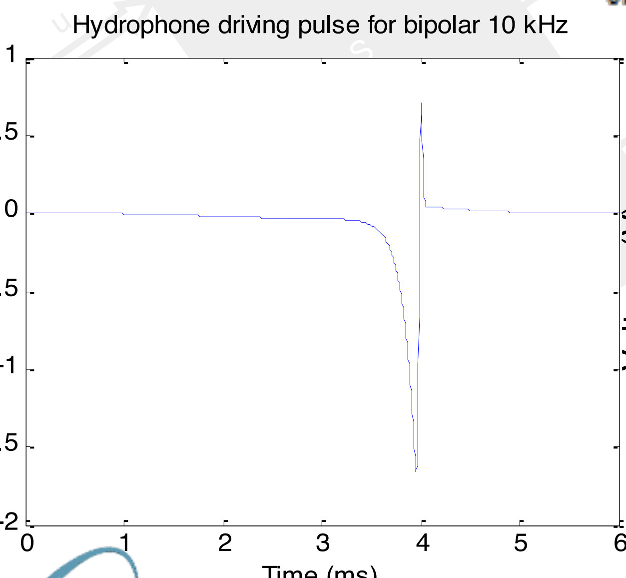
5^{th} Order Good enough for current hydrophones

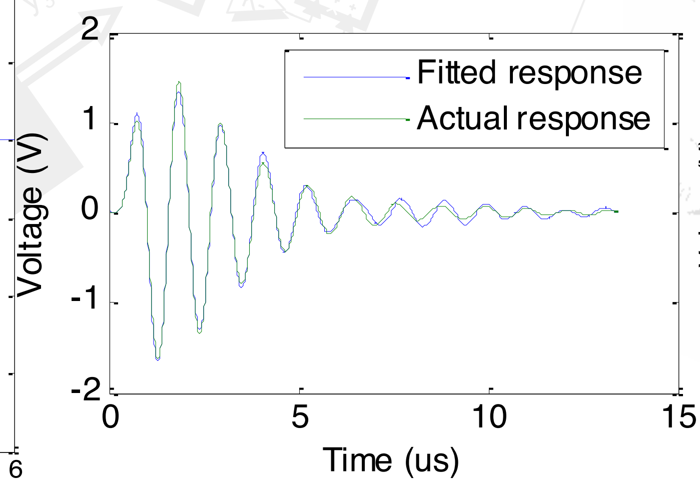


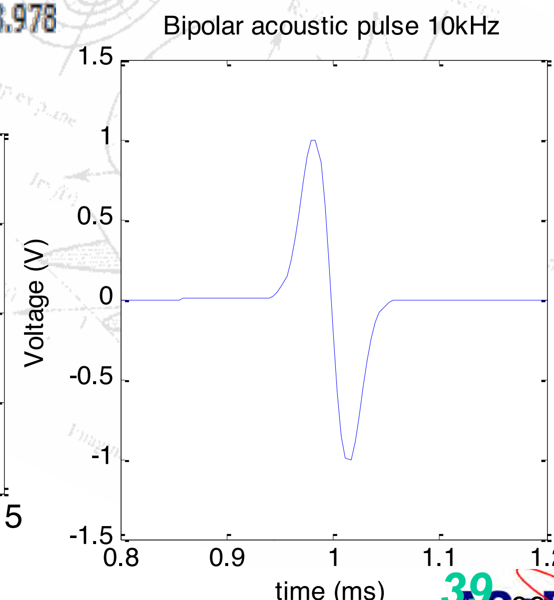




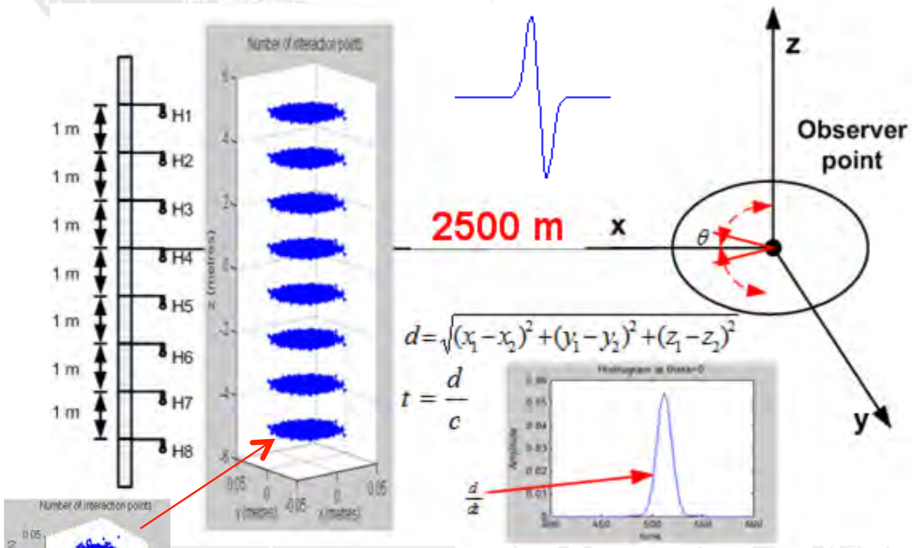
$$H'(s) = \frac{0.7153s^2 - 0.1063s + 0.003833}{0.0122s^5 + 0.04327s^4 + 0.941s^3 + 1.754s^2 + 16.65s + 8.978}$$

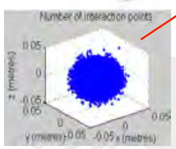


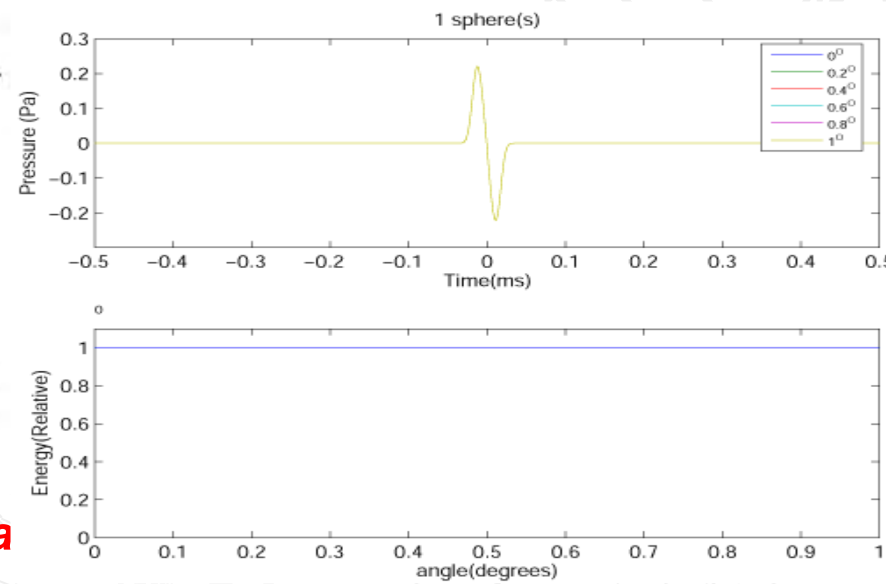




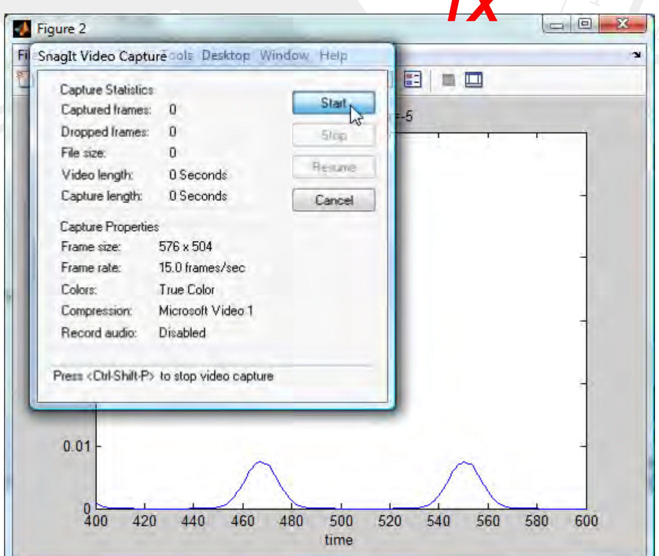
How Many Hydrophones needed for linear array?

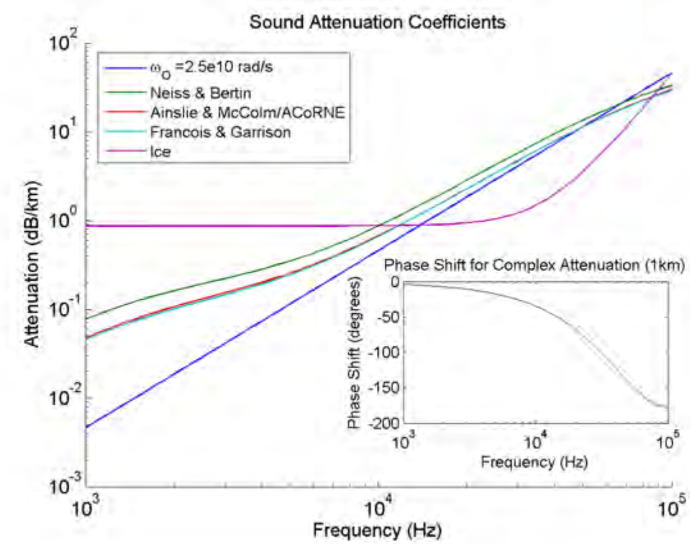




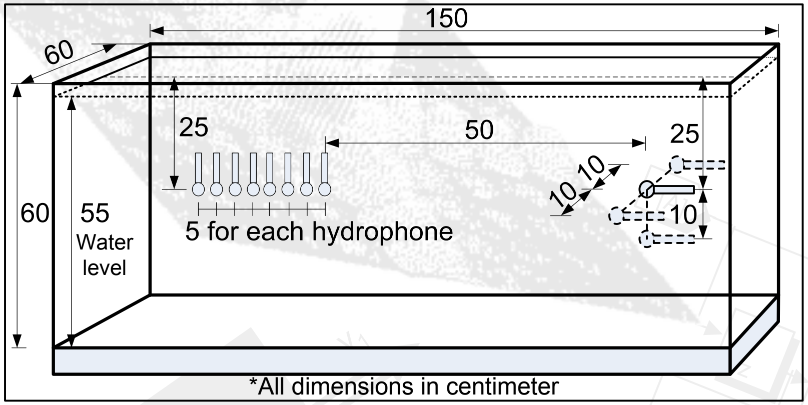


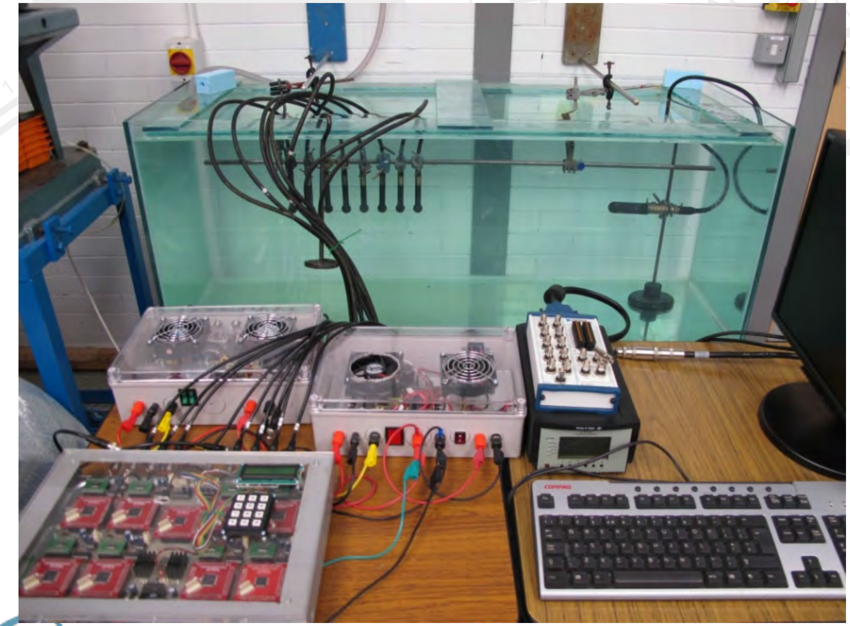
Simulation of 8 hydrophone a TX

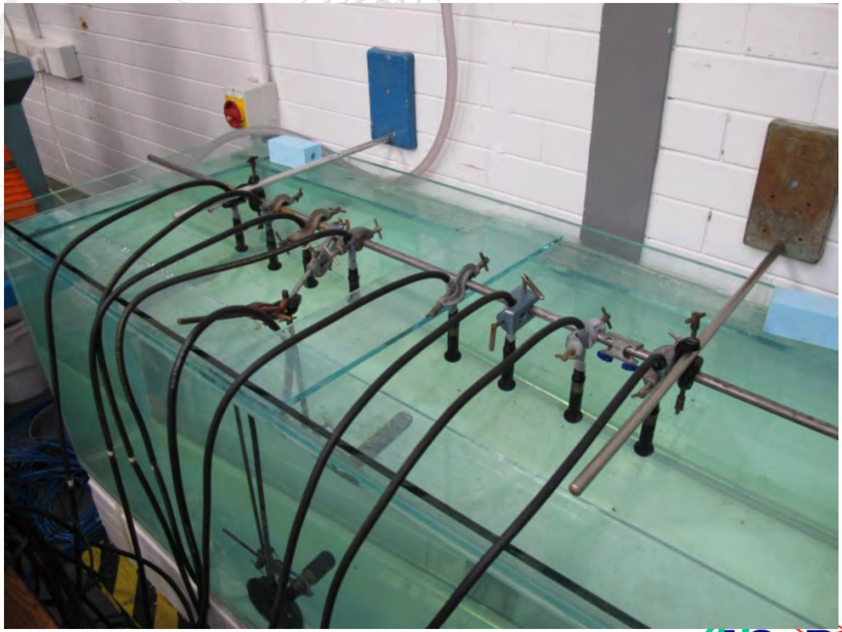


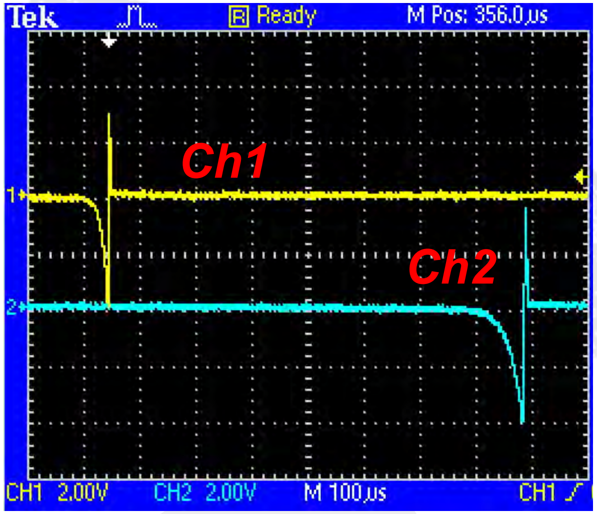


Laboratory at Northumbria University

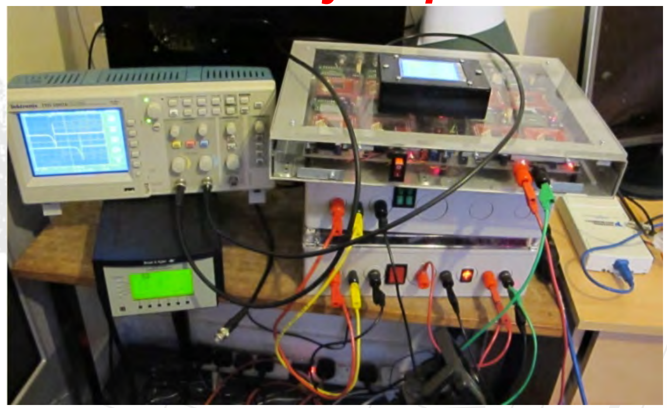


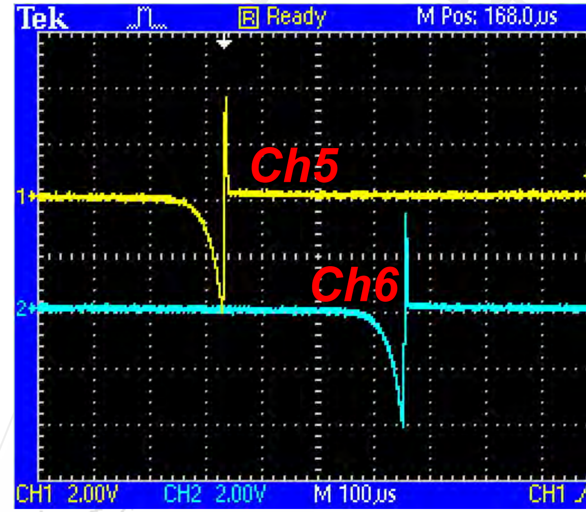


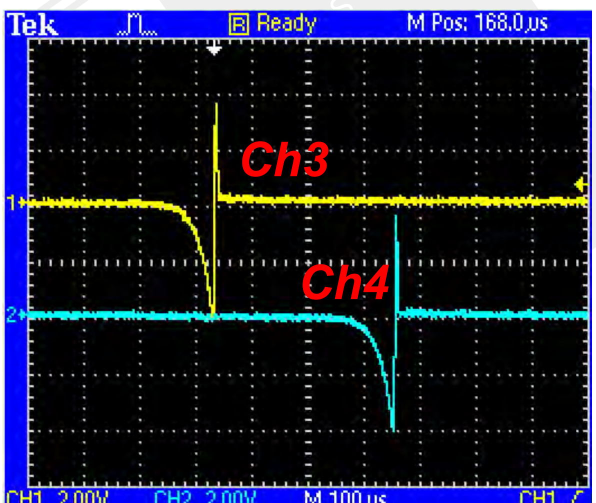


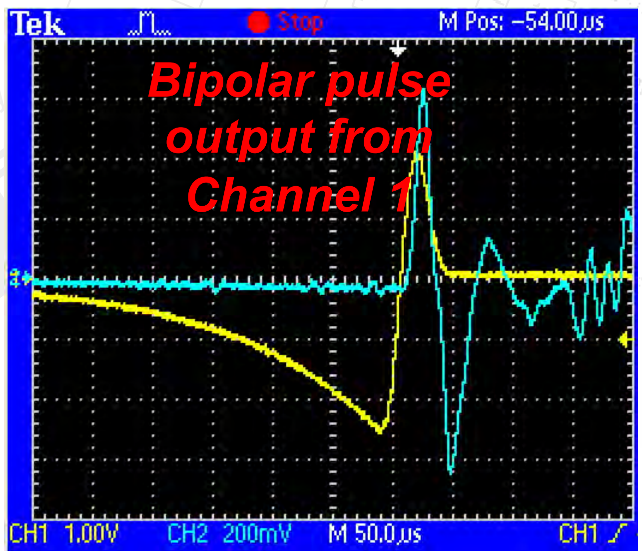


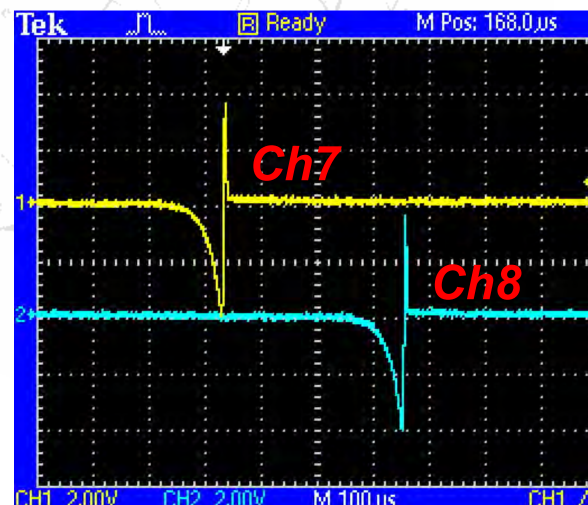
8 Channels hydrophone Tx







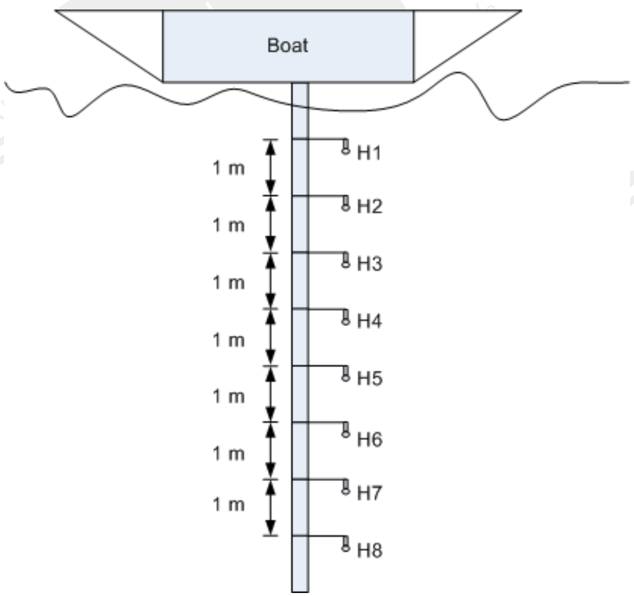


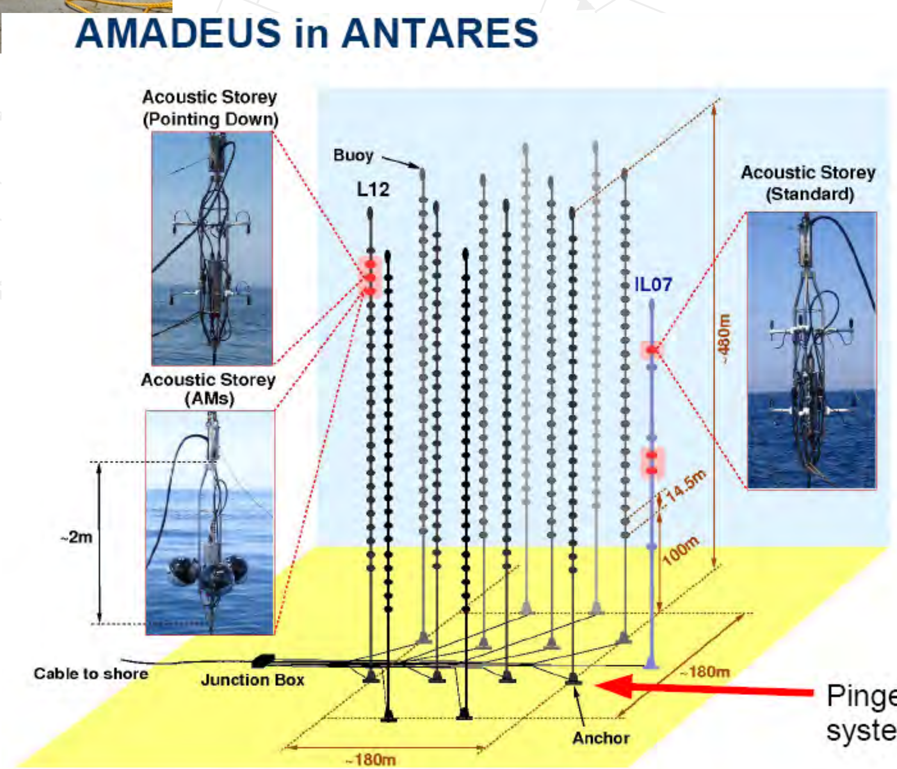


Deployment at ANTARES (France)

8 channel transmitter module

Deployment at ANTARES 17 September 2011





36 acoustic sensors on 6 storys

Pinger for positioning system


17/09/2011 08:11


17/09/2011 20:30

Digital Filters

northumbria UNIVERSITY Sean Danaher ACoRVE

The Digital Filter

The digital filter is simple! Based upon sampled sequences

$$a_0 y[n] + a_1 y[n-1] + a_2 y[n-2] + ... = b_0 u[n] + b_1 u[n-1] + b_2 u[n-2] + ...$$

Negative coefficients =>Causal

Simple moving average Filter

$$y[n] = \frac{1}{2}\left(x[n] + x[n-1]\right)$$

Crude low-pass Called FIR or MA

10	5
2	6
6	4
5	5.5
9	7
8	8.5
5	6.5
0	2.5
8	4
4	6

Perfect Integrator

$$y[n] = y[n-1] + x[n]$$

Called IIR or AR

6	6
8	14
9	23
7	30
2	32
4	36
9	45
9	54
4	58
9	67

The Z Transform and Sampled Signals

$$X(z)=\sum_{n=-\infty}^{n=\infty} x[n]z^{-n}$$

$$\text{if } x=1,3,4 \Rightarrow X(z)=1+3z^{-1}+4z^{-2}$$

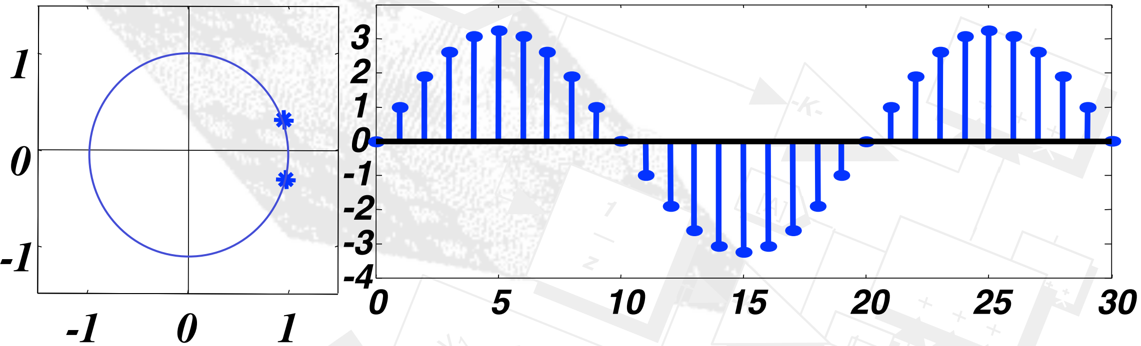
$$\text{if } x=1,a,a^2,\ldots \Rightarrow X(z)=\sum_{n=0}^{n=\infty}\left(az^{-1}\right)^n=\frac{1}{1-az^{-1}}$$

Geometric sequences are of prime interest because

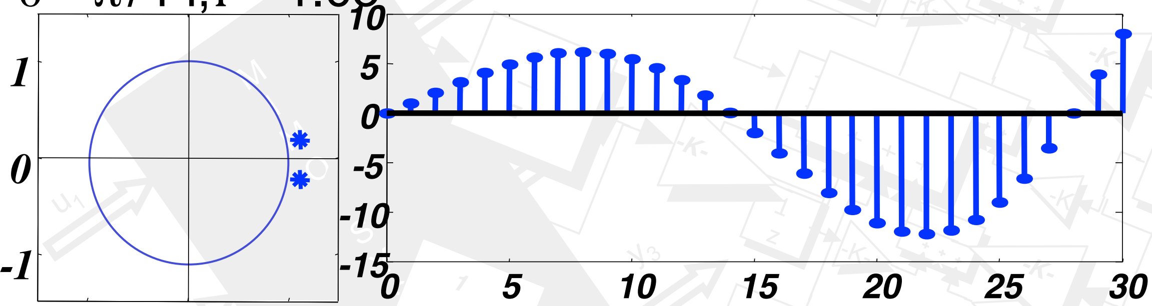
$u(t)=e^{i\omega t} \Rightarrow u[t]=e^{i\Omega nt}$ ***is a geometric sequence***

A few z transforms

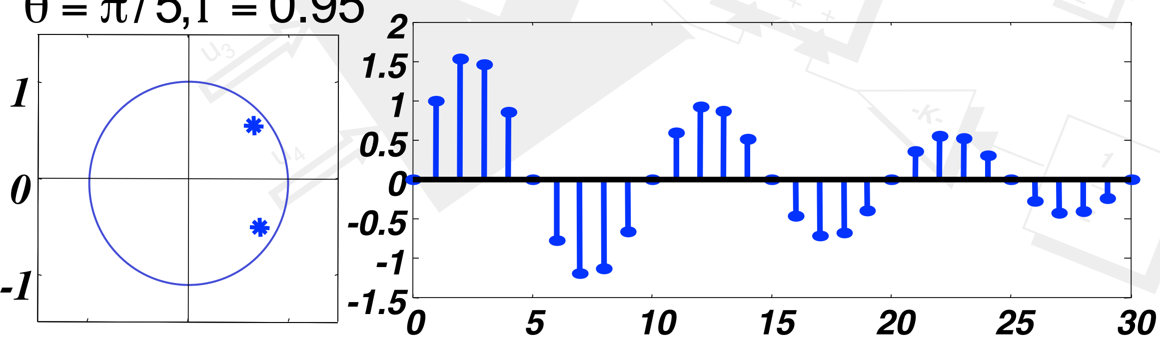
$\theta = \pi/10, \lambda = 1$



$\theta = \pi/14, \lambda = 1.05$



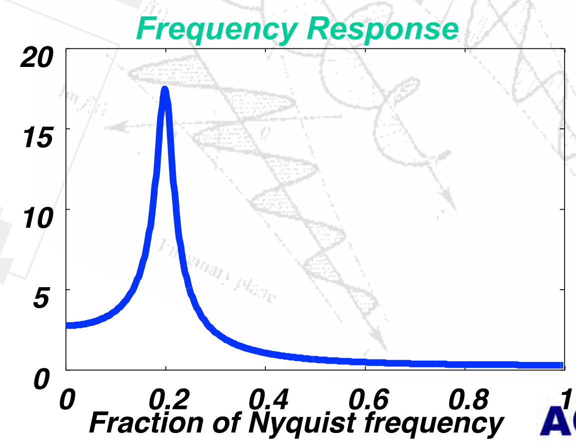
$\theta = \pi/5, \lambda = 0.95$



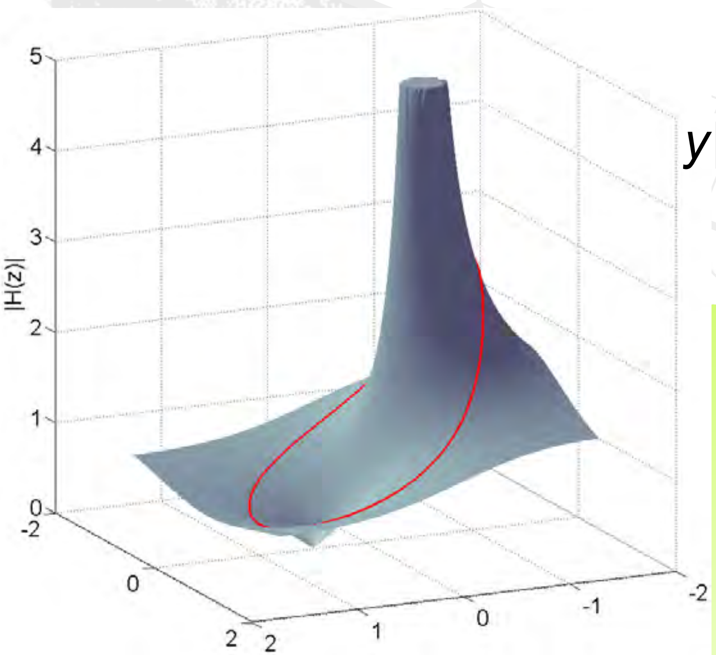
$$h(z) = \frac{1}{z^{-2} - 2\cos\left(\frac{\pi}{10}\right) z^{-1} + 1}$$

$$y[n] = x[n] + 2\cos\left(\frac{\pi}{10}\right) y[n-1] - y[n-2]$$

Poles must be inside the unit circle for stability

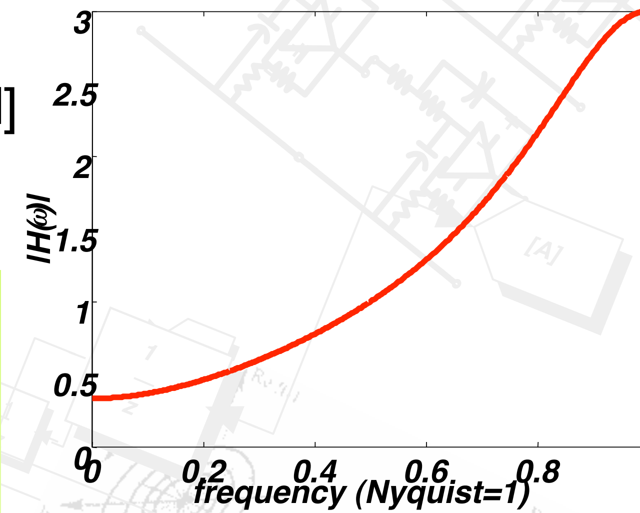


Simple Transfer Function

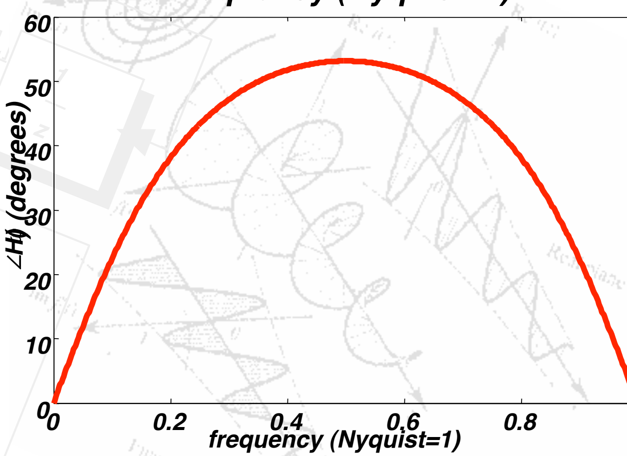


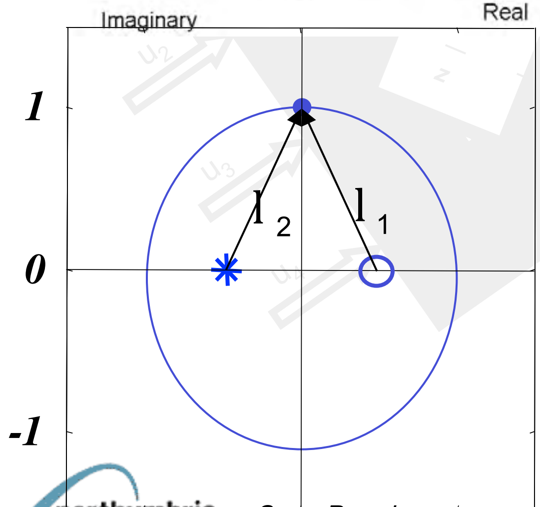
$$h(z) = \frac{z - 0.5}{z + 0.5}$$

$$y[n] = x[n] - 0.5x[n-1] - 0.5y[n-1]$$



Frequency response simply determined by running around the unit circle
π Corresponds to the Nyquist Frequency





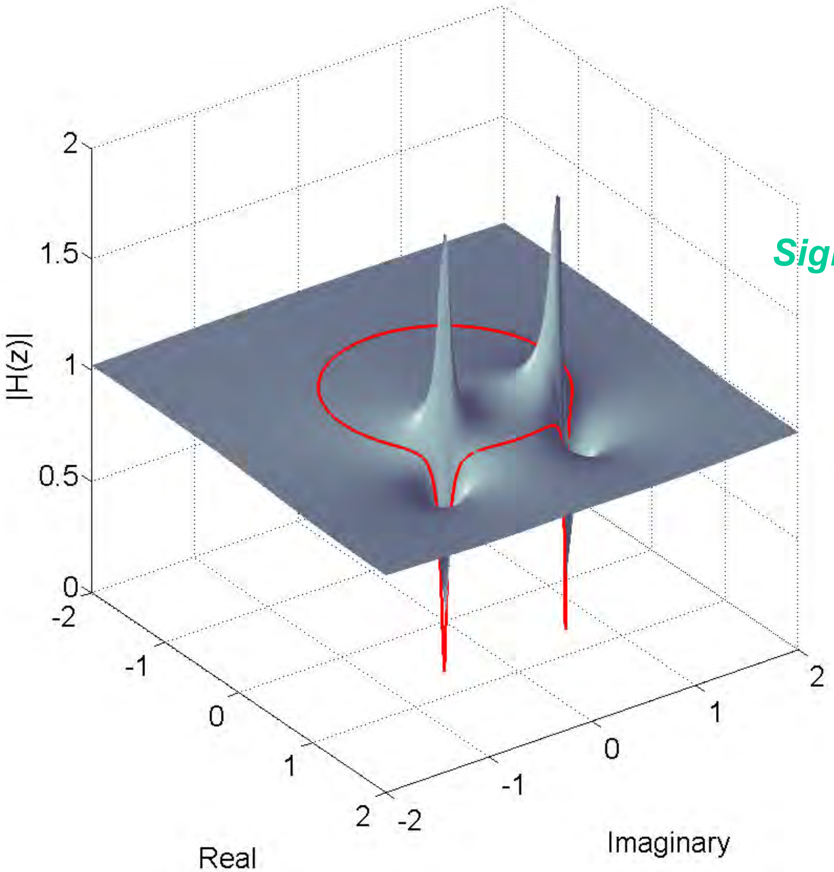
$$|H(z)| = \frac{l_1}{l_2}$$

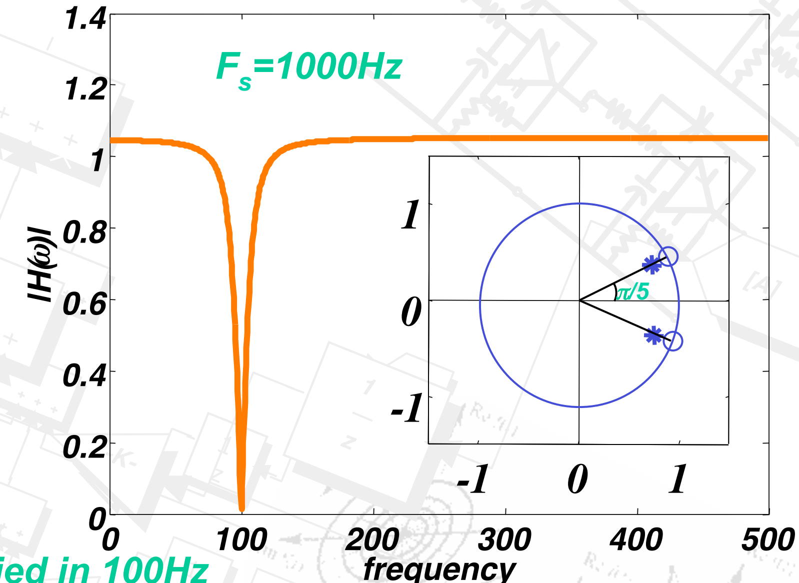
$$\angle H(z) = \angle\theta_1 - \angle\theta_2$$

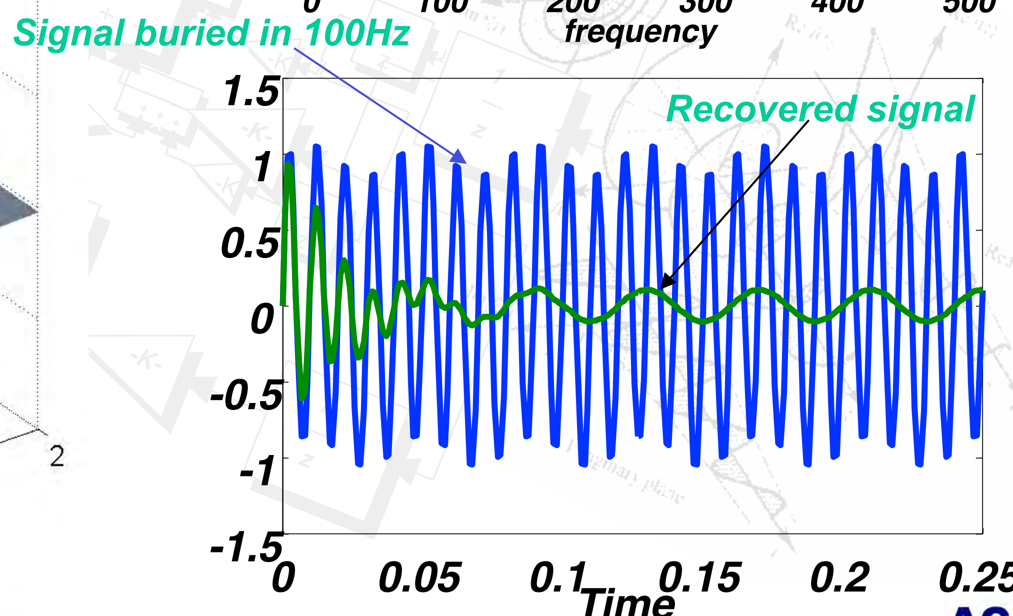
Notch Filter

$$h(z) = \frac{(z - e^{i\frac{\pi}{5}})(z - e^{-i\frac{\pi}{5}})}{(z - 0.95e^{i\frac{\pi}{5}})(z - 0.95e^{-i\frac{\pi}{5}})}$$

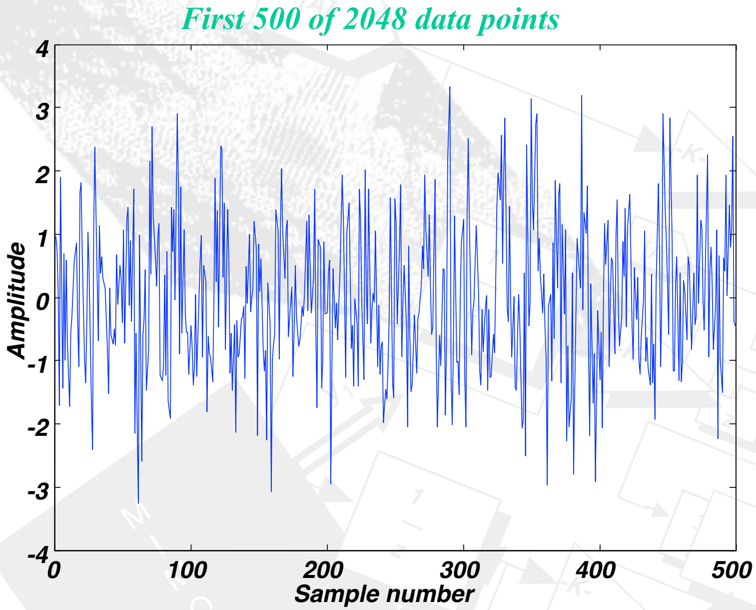
$$y[n]=x[n]-1.6180x[n-1]+x[n-2]$$
$$+1.5371y[n-1]-0.9025y[n-2]$$





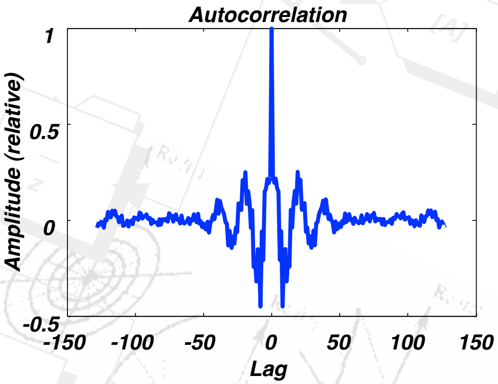


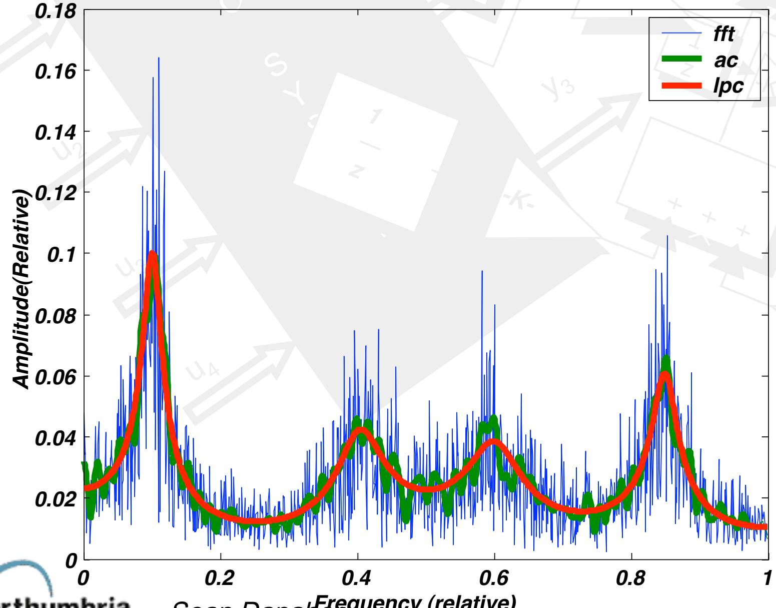
Spectral Analysis



Using the Fourier Transform is seldom the best way to get a spectrum. Normally methods based around
- Autocorrelation (AC)
- Linear Prediction are used

AC method
Use FT of AC
To get PSD





Linear Prediction

$$a_0 y[n] + a_1 y[n-1] + a_2 y[n-2] .... = e[n]$$

All pole filter driven by white noise. Need to choose the order with care. But can now reproduce the spectrum

Compression and feature extraction

FFT does not compress
Often need to extract features in a few parameters

DSP people use FIR and IIR filters and system identification
Statisticians use MA, AR, ARMA, ARMAX models
Human speech compression possibly the most advanced form of signal processing
Historically these models are called vocoders
Very possibly these techniques can be applied to marine mammals

Modern mobile standards such as GSM are based on CELP – Code Excited Linear Prediction
The first CELP algorithm around '94 took c 24 hours to code 5 mins of speech
Initial development done by the US military for secure communiaction
Rapidly advanced to work in real time. Mobile telephone companies poured a lot of money into algorithm development.

Human Speech

Vocal tract acts like a resonator. There are typically a number of resonances called formants
We solve the equation

$$y[n] = a_1 y[n-1] + a_2 y[n-2] + a_3 y[n-3] + \ldots a_k y[n-k] + e[n]$$

This is an IIR filter, an ARX process or a linear predictor
LPC10 has 10 coefficients (k=10)

Hearing LPC in action

- 8 bit 8kHz
- Split speech into frame 240 samples long
- Use LPC10 to estimate spectrum
- Reconstruct

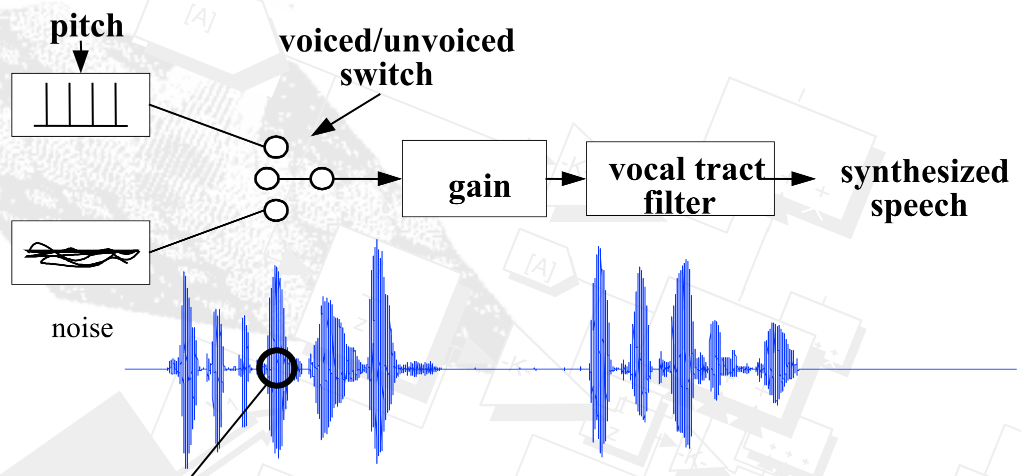
64k

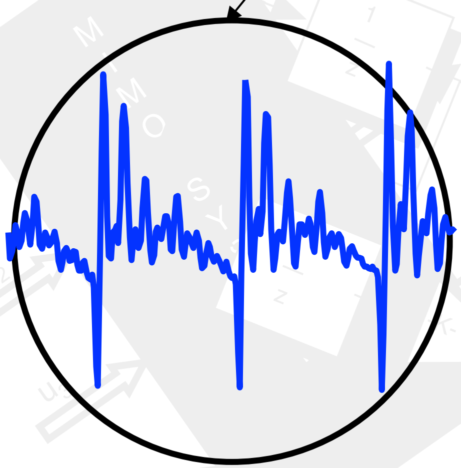
933

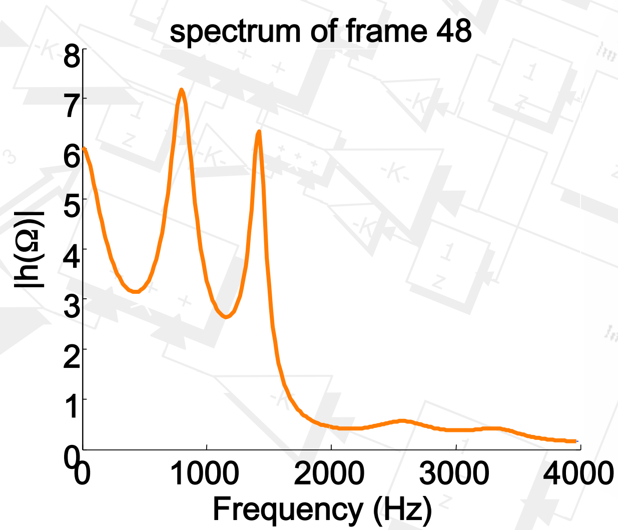
933

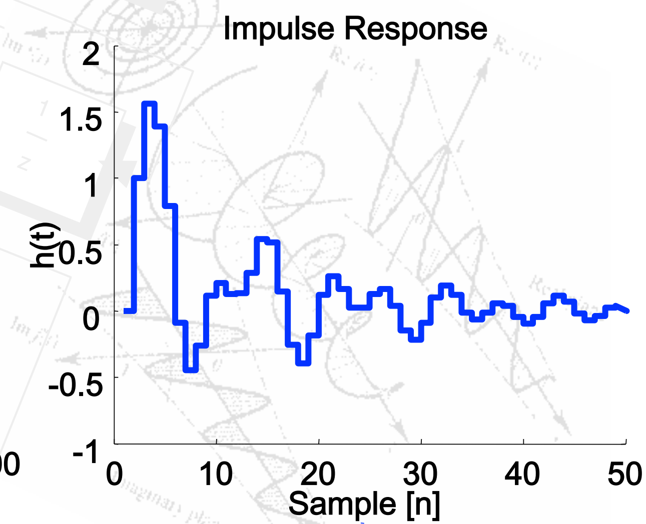
1.2k

2.4k









SVD

Singular Value Decomposition

We can get an approximation of the original data by setting the L values to zero below a certain threshold

Similar techniques are used in statistics CVA, PCA and Factor Analysis

Based on Eigenvector Techniques

Good SVD algorithms exist in ROOT and MATLAB

Decompose the Data matrix into 3 matrices.

When multiplied we get back the original data.

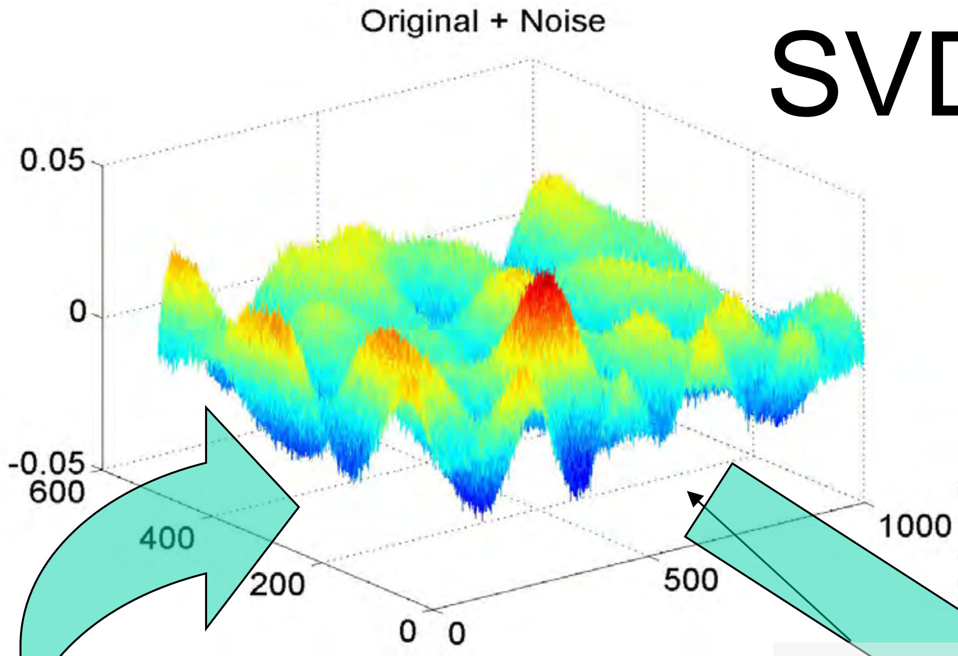
W and V are unitary.

L contains the contribution from each of the eigenvectors in descending order along main diagonal.

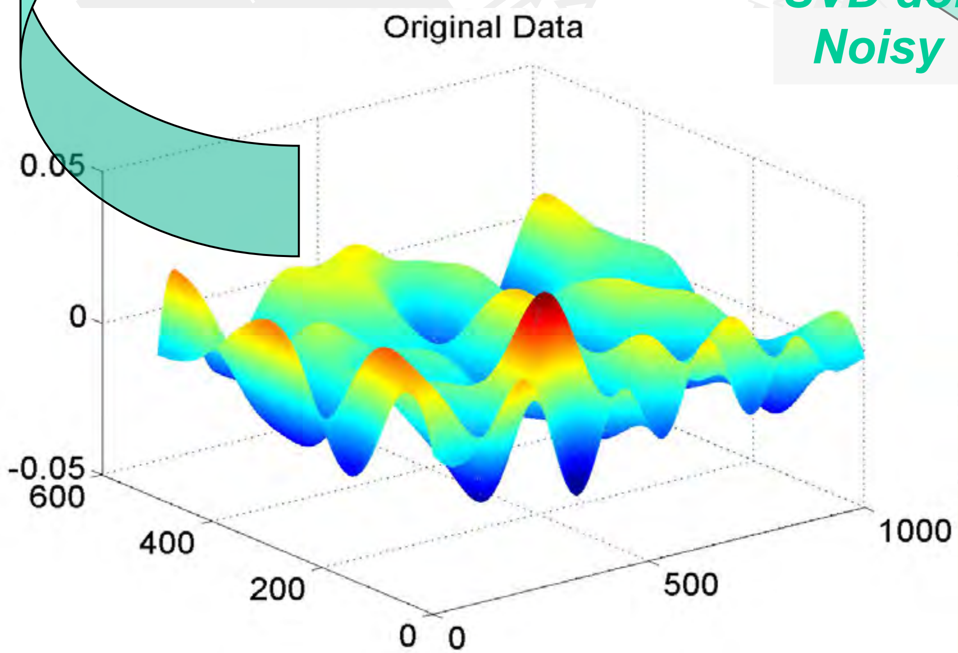
SVD II

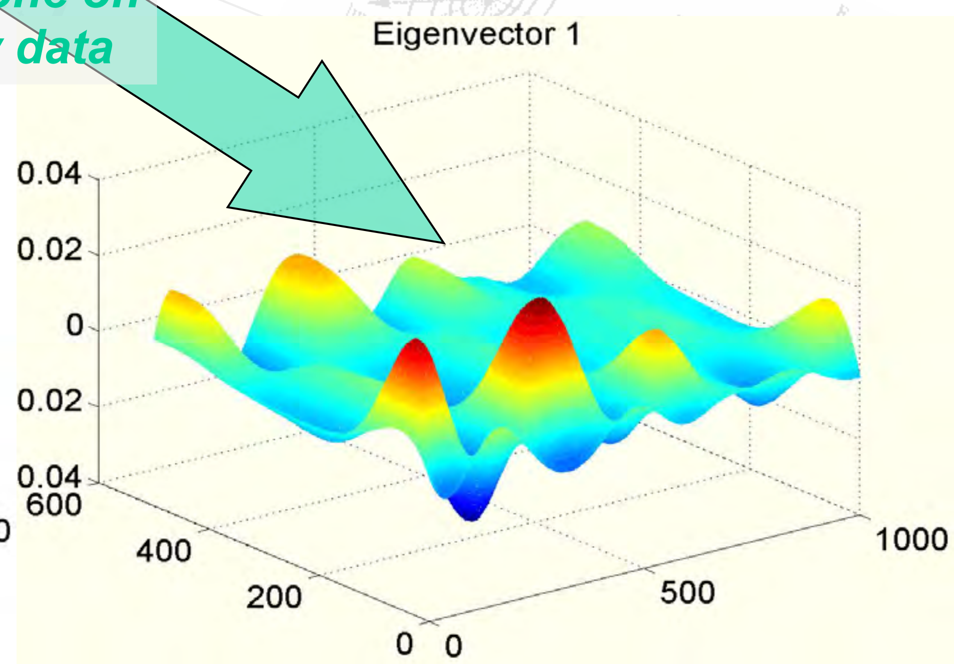
Somewhere between vector five and seven we get the "Best" representation of the data

- *Better than the observation as noise filtered out*
- *Highly compressed (only 1-2% of original size e.g. 6/500x1.5)*
- *Have basis vectors for data so can produce "similar" data+++*
- *Use it before going to classification or clustering algorithms*



SVD done on Noisy data





Conclusions

- Only Scratched the surface
- Questions