DISCRETISATION OF THE HARMONIC OSCILLATOR

Maarten van Walstijn

Aims

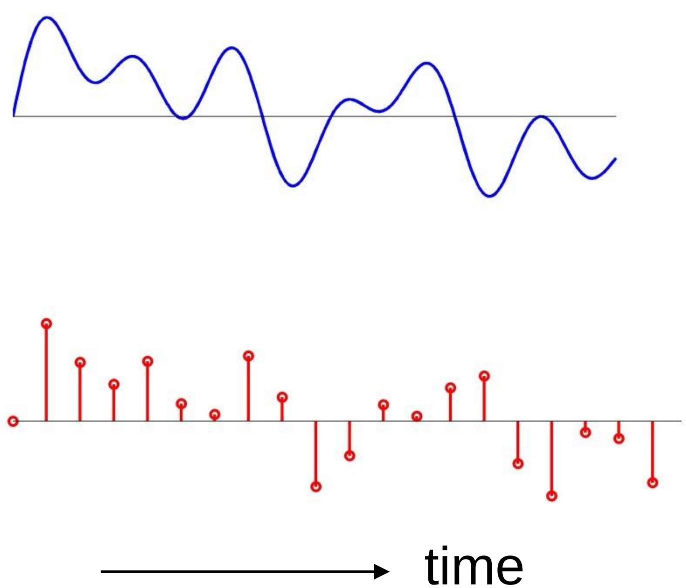
- Introduce the basic principles and methods used in discretisation of continuous-time systems.
- Show how to apply basic discretisation methods to simple vibrating systems such as the harmonic oscillator.
- Introduce methods for analysis of stability and discretisation artefacts.
- Explore the analogy with filter theory.

OVERVIEW

- Discrete-Time Modelling
- Finite-Difference (FD) Schemes
- FD simulation of a Harmonic Oscillator
- Stability Analysis
- Discretisation Artefacts
- Impulse-Invariant Method
- Frequency Response

Discrete-Time Modelling

Discretisation is a procedure that transforms a continuous-time (analog) model of a vibrating system into a discrete-time (digital) model.

	single point mass motion:	signal:	model:
continuous-time domain		*analog signal*	*analog model*
		sampling	*discretisation*
discrete-time domain		*digital signal*	*digital model*

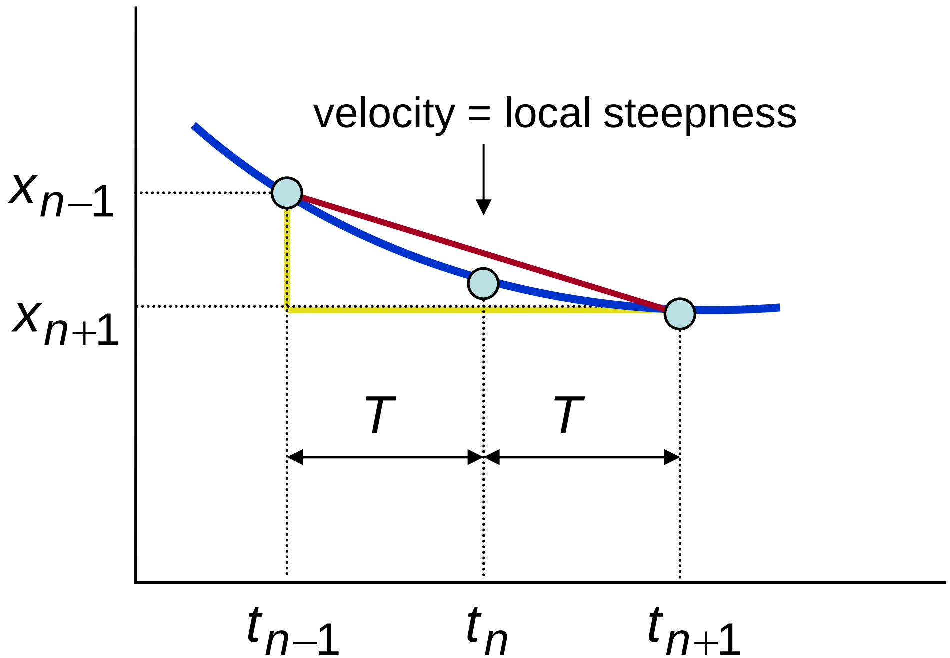
Finite Difference Schemes

Approximating derivatives with differences

Example: centered-difference scheme for digital approximation of velocity:

$$x_n = \text{position at time } t_n = nT$$

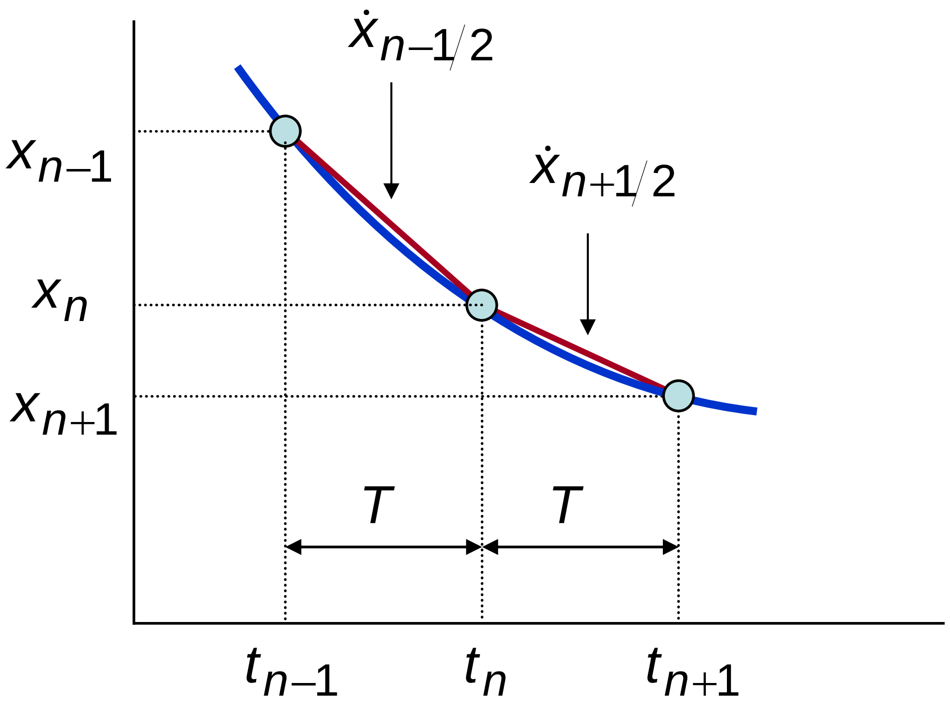
$$\dot{x}_n = \left[\frac{dx}{dt}\right]_n = \text{velocity at time } t_n = nT$$



$$\dot{x}_n \approx \frac{x_{n+1} - x_{n-1}}{2T}$$

Finite Difference Schemes (cont.)

acceleration: $\ddot{x}_n = \left[\dfrac{d^2x}{dt^2}\right]_n$ = accel. at time $t_n = \mathrm{nT}$



$$\ddot{x}_n \approx \frac{\dot{x}_{n+1/2} - \dot{x}_{n-1/2}}{T}$$

$$\approx \frac{\dfrac{x_{n+1} - x_n}{T} - \dfrac{x_n - x_{n-1}}{T}}{T}$$

$$= \frac{x_{n+1} - 2x_n + x_{n-1}}{T^2}$$

FD Simulation of a Driven Harmonic Oscillator

Derivation of the discrete-time model:

Equation of motion:	$m\,\ddot{x} + r\,\dot{x} + k\,x = F \;\Leftrightarrow$ $\ddot{x} + 2\alpha\,\dot{x} + \omega_0^2\,x = \dfrac{F}{m}$
centered-difference scheme:	$\begin{cases} \dot{x}_n \approx \dfrac{x_{n+1} - x_{n-1}}{2T} \\ \ddot{x}_n = \dfrac{x_{n+1} - 2x_n + x_{n-1}}{T^2} \end{cases}$ $\quad (\Delta t \equiv T)$

Substitution of the FD scheme into the equation of motion yields:

$$x_{n+1} - 2x_n + x_{n-1} + \alpha T\left(x_{n+1} - x_{n-1}\right) + \omega_0^2 T^2 x_n = \frac{T^2 F_n}{m}$$

FD Simulation of a Driven Harmonic Oscillator (cont.)

Collect terms in x:

$$(1+\alpha T)x_{n+1} + (\omega_0^2 T^2 - 2)x_n + (1-\alpha T)x_{n-1} = \frac{T^2 F_n}{m}$$

Write the 'newest' value explicit:

$$x_{n+1} = \frac{-(\omega_0^2 T^2 - 2)x_n + (\alpha T - 1)x_{n-1} + \dfrac{F_n}{mT^2}}{(\alpha T + 1)}$$

$\Leftrightarrow$

$$x_{n+1} = b_1 F_n - a_1 x_n - a_2 x_{n-1}$$

difference equation

where

coefficients

$$a_1 = \frac{\omega_0^2 T^2 - 2}{1+\alpha T}$$

$$a_2 = \frac{1-\alpha T}{1+\alpha T}$$

$$b_1 = \frac{\left(T^2 / m\right)}{1+\alpha T}$$

FD Simulation of a Driven Harmonic Oscillator (cont.)

Difference equation computes the new value from a linear combination of the past two values and the last force-input value.

$$x_{n+1} = b_1 F_n - a_1 x_n - a_2 x_{n-1}$$

new value | last force input value | last value | before-last value

The discrete-time output of the model can be computed iteratively:

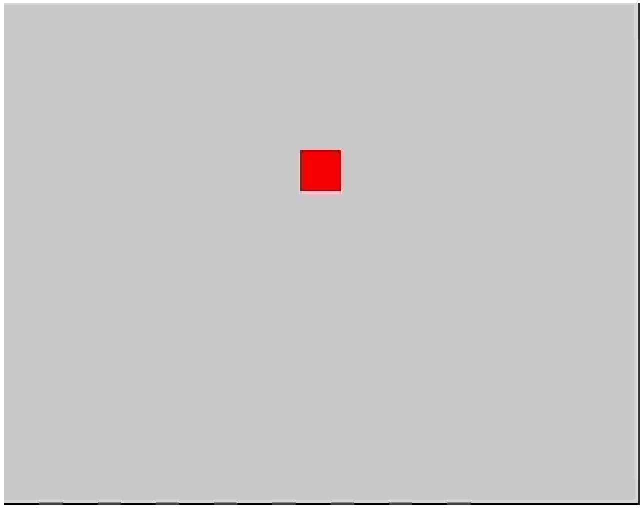
1. set $x_n = x_{n-1} = 0$
2. for $n = 0$ to $N, do:$

$$x_{n+1} = b_1 F_n - a_1 x_n - a_2 x_{n-1}$$
$$x_{n-1} = x_n$$
$$x_n = x_{n+1}$$

Example Matlab Code on Website: **harmosc.m**

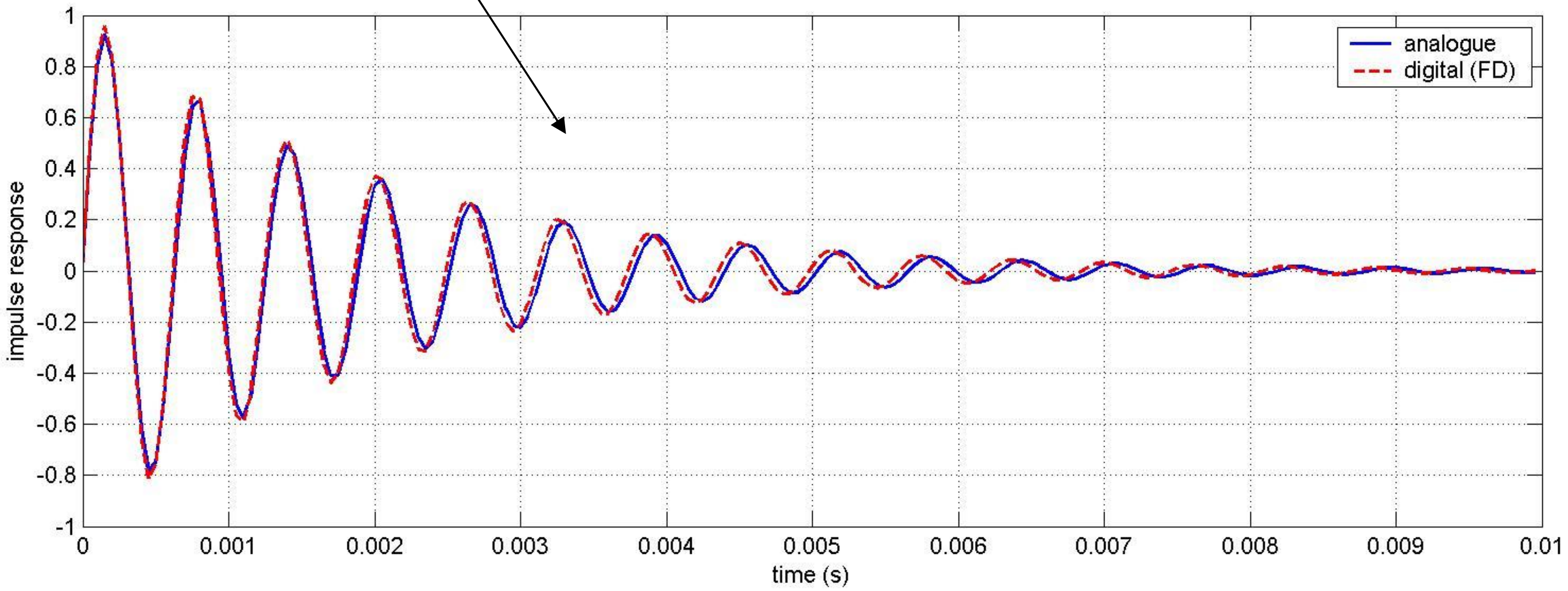
FD Simulation of a Driven Harmonic Oscillator (cont.)

Compute impulse response
(i.e. F_n=*0* except F_0=*1*):



Fs = 20000, m=0.0001, k =10000, r=0.1

Simulation works, but
not without artefacts?



Stability & Oscillation

Is simulation always stable? Do we always obtain an oscillatory solution? If not, under which circumstances?

Assume sinusoidal, exponentially decaying solution for free vibration of the discrete-time model:

$$x(t) = e^{st} \Rightarrow \quad x_n = e^{snT} \qquad\qquad (s = j\omega_d - \alpha_d)$$

Thus we may write 'new value' in terms of 'last value'

$$x_{n+1} = e^{s(n+1)T} = e^{sT} e^{snT} = z\, x_n \quad (z = e^{sT})$$

Where z is the **amplification factor**.

For the discrete-time solution to be oscillatory and stable we have:

Condition 1: $im\{z\} \neq 0$ $\longrightarrow$ otherwise, no oscillation

Condition 2: $|z| \leq 1$ $\longrightarrow$ otherwise, growing rather than decaying oscillation

Stability & Oscillation (cont.)

Substitution of the relationships into the difference equation

$$x_{n+1} = b_1 F_n - a_1 x_n - a_2 x_{n-1} \quad \Longleftarrow \quad \begin{cases} x_n = z\, x_{n-1} \\ x_{n+1} = z^2 x_{n-1} \end{cases}$$

gives (for stability analysis we may assume the driving force = 0):

$$z^2 x_{n-1} = -a_1 z x_{n-1} - a_2 x_{n-1}$$

$$\Leftrightarrow$$

(must be negative)

$$x_{n-1}\left(z^2 + a_1 z + a_2\right) = 0 \quad \Rightarrow \quad z = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2}$$

Since z is complex, we have:

$$z = \frac{-a_1 \pm j\sqrt{D}}{2} \qquad \left(\mathrm{D} = 4a_2 - a_1^2 > 0\right)$$

Stability & Oscillation (cont.)

Two complex solutions that are each others conjugate, thus have the same modulus, for which we have:

$$|z|^2 = \frac{1}{4}\left(a_1 - j\sqrt{D}\right)\left(a_1 + j\sqrt{D}\right) = \frac{1}{4}\left(a_1^2 + D\right)$$

$$= \frac{1}{4}\left(a_1^2 + 4a_2 - a_1^2\right) = a_2$$

Impose stability conditions:

Condition 1: $4a_2 - a_1^2 > 0$

Condition 2: $|z| < 1 \;\Leftrightarrow\; |z|^2 < 1 \;\Rightarrow\; a_2 < 1$

Stability and Oscillation (cont.)

For the employed FD scheme we have

$$a_1 = \frac{\omega_0^2 T^2 - 2}{1+\alpha T} \qquad\qquad a_2 = \frac{1-\alpha T}{1+\alpha T}$$

The second condition leads to:

$$\frac{1-\alpha T}{1+\alpha T} < 1 \quad \Rightarrow \quad \alpha T > 0$$

this is true for all possible sample rates

The first condition (oscillatory) becomes:

$$4\left(\frac{1-\alpha T}{1+\alpha T}\right) - \left(\frac{\omega_0^2 T^2 - 2}{1+\alpha T}\right)^2 > 0 \quad \Leftrightarrow \quad 4\left(1-\alpha^2 T^2\right) > \left(\omega_0^2 T^2 - 2\right)^2$$

Stability and Oscillation (cont.)

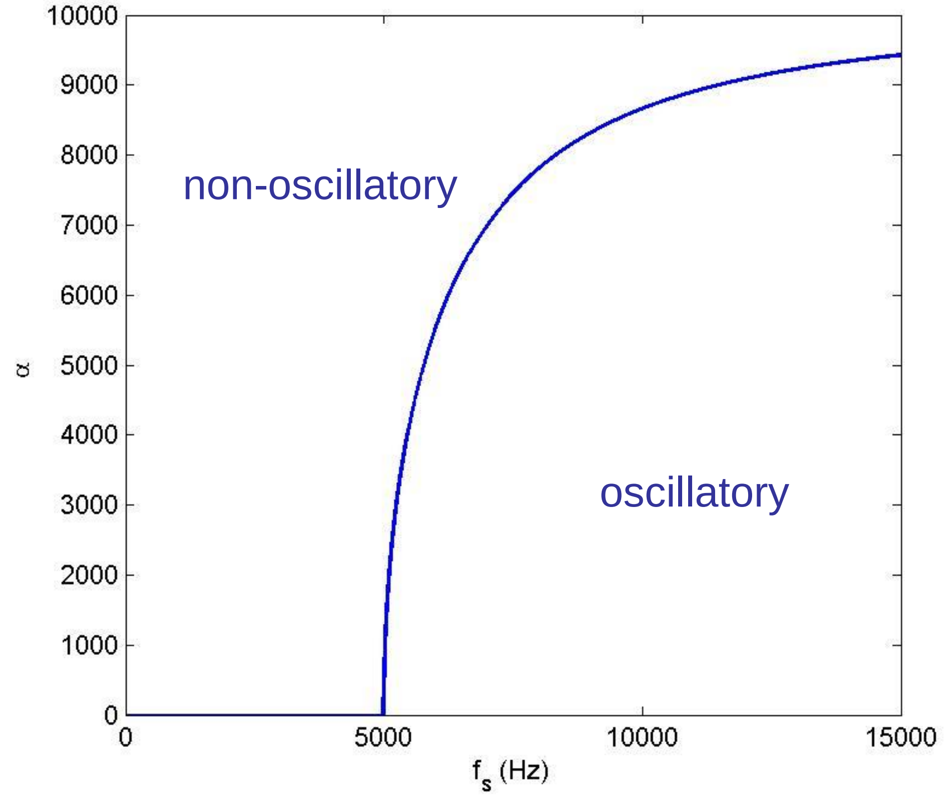
Which leads to:

$$T < \frac{2\sqrt{\omega_0^2 - \alpha^2}}{\omega_0^2}$$

Thus the sample rate must be chosen such that:

$$f_s > \frac{\omega_0^2}{2\sqrt{\omega_0^2 - \alpha^2}}$$

$\omega_0 = 10000$



Discrete-Time Resonance Frequency

For free vibration, the discrete-time system oscillates at its (own) resonance frequency. In that case we have:

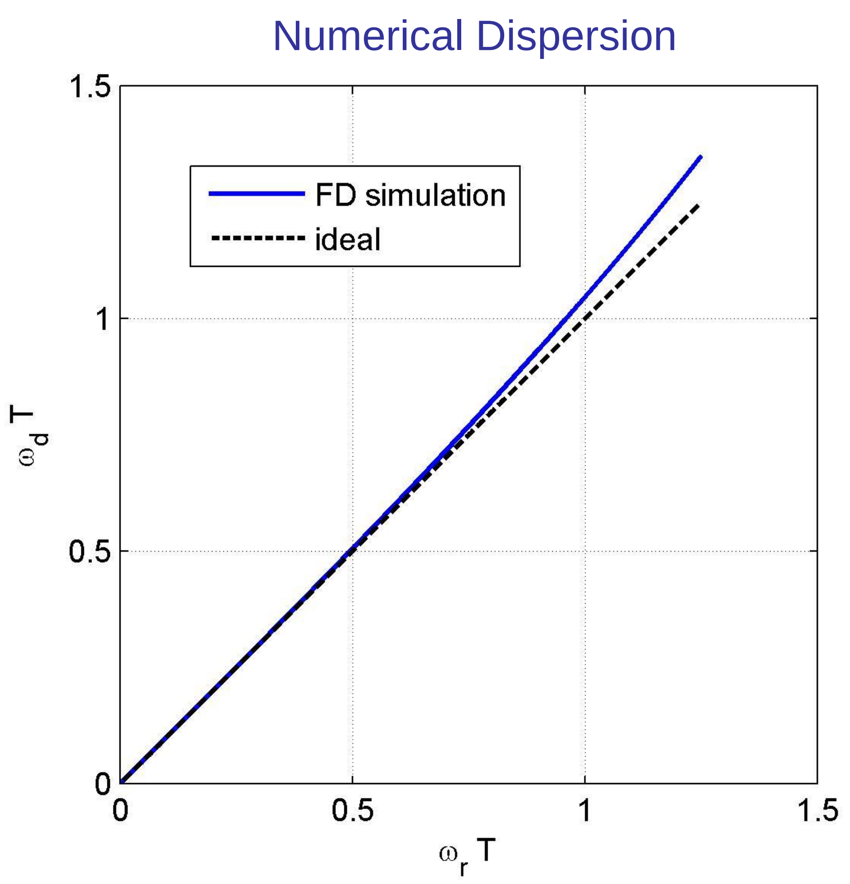
$$z = e^{sT} = e^{(j\omega_d - \alpha_d)T} = \frac{-a_1 \pm j\sqrt{D}}{2}$$

Thus the discrete-time resonance frequency is:

$$\omega_d = \frac{\angle z}{T} = f_s \tan^{-1}\left(\frac{im\{z\}}{re\{z\}}\right)$$

$$= f_s \tan^{-1}\left(\sqrt{\frac{4a_2}{a_1^2} - 1}\right)$$

Example plot (see right).



Discrete-Time Damping

For the discrete-time system in free vibration, we have:

$$z = e^{(j\omega_d - \alpha_d)T}$$

$$|z|^2 = zz^* = e^{-2\alpha_d T} = a_2$$

Thus we can write

$$-2T\alpha_d = \log(a_2)$$

Hence the discrete-time attenuation coefficient is:

$$\alpha_d = -\frac{\log(a_2)}{2T}$$

In other words, the discretisation process can also change the damping properties: *numerical attenuation.*

Discretisation without Artefacts at Resonance

Can we find alternative coefficients a_1 and a_2, for which the difference equation which produces free vibration with the same resonance frequency and damping as the continuous-time model? This would mean:

$$z^2 + a_1 z + a_2 = 0 \qquad \text{with} \quad z = e^{(j\omega_r - \alpha)T}$$

Which can be written

$$e^{(j\omega_r - \alpha)T} + a_1 + a_2 e^{(-j\omega_r + \alpha)T} = 0 \qquad \left[\text{set } R = e^{-\alpha T} + \text{use Euler's rule}\right]$$

$$\Leftrightarrow$$

$$R\left[\cos(\omega_r T) + j\sin(\omega_r T)\ \right] + a_1 + a_2 R^{-1}\left[\cos(\omega_r T) - j\sin(\omega_r T)\ \right] = 0$$

Both real and imaginary parts equal should equal zero, thus:

$$\begin{cases} R\sin(\omega_r T) - a_2 R^{-1}\sin(\omega_r T) = 0 \\ R\cos(\omega_r T) + a_1 + a_2 R^{-1}\cos(\omega_r T) = 0 \end{cases}$$

Discretisation without Artefacts at Resonance (cont.)

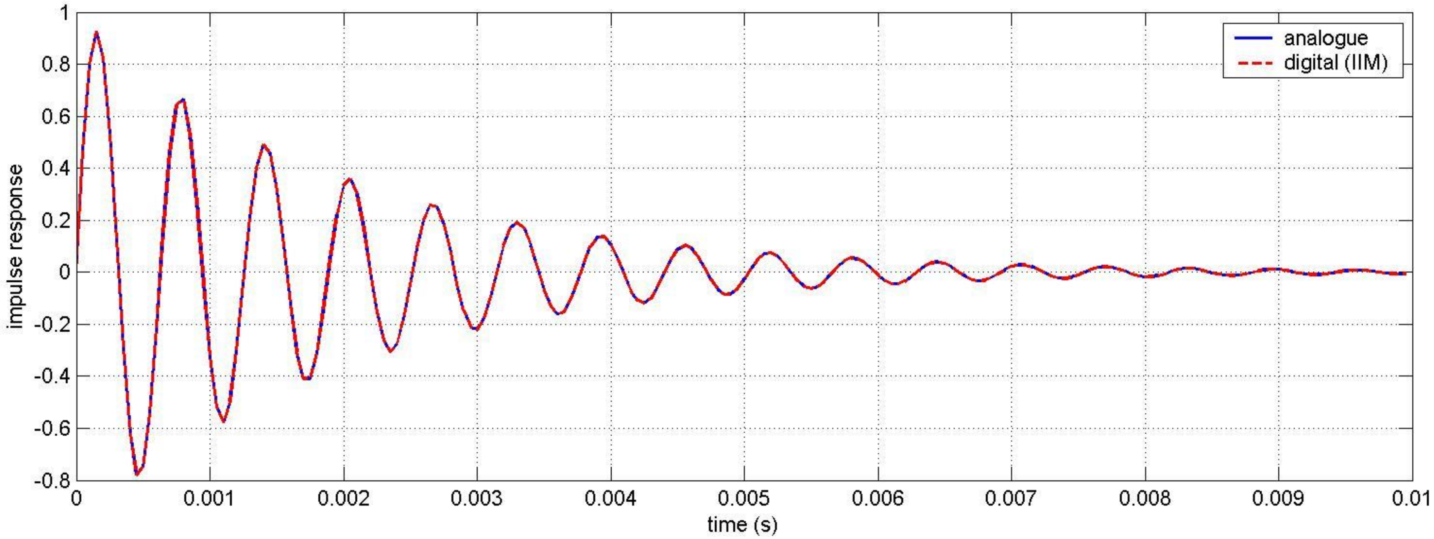
This leads to:

$$\begin{cases} a_2 = R^2 \\ a_1 = -2R\cos(\omega_r T) \end{cases}$$

For ‘sampled’ impulse response, set:

$$b_1 = \left(\frac{RT}{M\omega_r}\right)\sin(\omega_r T)$$

This is referred to, in digital filter design context, as as the Impulse-Invariant Method (IIM).



Discrete-Time Frequency Response

How to compute the frequency response of the discrete-time system?

Start with difference equation:

$$x_{n+1} = b_1 F_n - a_1 x_n - a_2 x_{n-1}$$

Linear system, thus sinusoidal excitation gives sinusoidal output:

$$F_n = Ae^{j\omega nT}$$

$$x_n = Be^{j\omega nT}$$

Substitute into difference equation:

$$Be^{j\omega(n+1)T} = b_1 Ae^{j\omega nT} - a_1 Be^{j\omega nT} - a_2 Be^{j\omega(n-1)T}$$

or

$$Be^{j\omega nT} e^{j\omega T} + a_1 Be^{j\omega nT} + a_2 Be^{j\omega nT} e^{-j\omega T} = b_1 Ae^{j\omega nT}$$

Discrete-Time Frequency Response (cont.)

This leads to

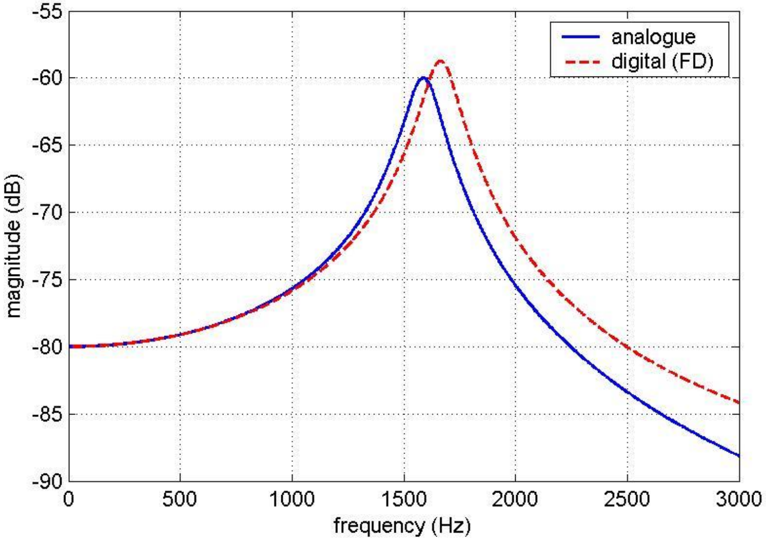
$$B\left(e^{j\omega T} + a_1 + a_2 e^{-j\omega T}\right) = b_1 A$$

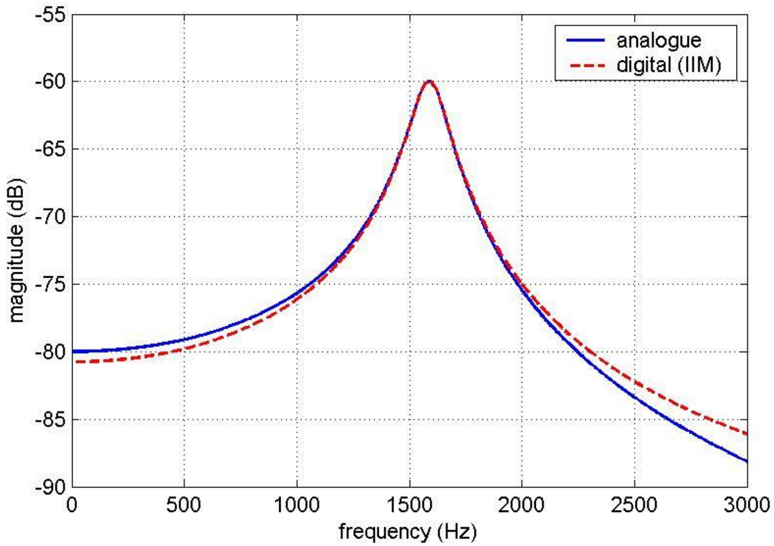
Thus the frequency response is:

$$H_d(\omega) = \frac{x_n}{F_n} = \frac{B}{A}$$

$$= \frac{b_1}{e^{j\omega T} + a_1 + a_2 e^{-j\omega T}}$$

Example plot (f_s=1000Hz), comparing analogue to digital response:





Digital Filter Transfer Function (1)

If we assume exponentially decaying solution:

$$F_n = Ae^{snT} \qquad\qquad (s = j\omega - \alpha)$$

$$x_n = Be^{snT}$$

And follow through the same procedure as for the frequency response, we obtain:

$$H_d(s) = \frac{b_1}{e^{sT} + a_1 + a_2 e^{-sT}} = \frac{b_1 e^{-sT}}{1 + a_1 e^{-sT} + a_2 e^{-2sT}}$$

If we then use again $z = e^{j\omega T}$, we obtain the **digital filter transfer function** of the system:

$$H_d(z) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

The frequency response is thus the transfer function evaluated at $s = j\omega$

Digital Filter Transfer Function (2)

General form for digital transfer function of order:

$$H_d(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} \ldots + b_n z^{-n} \ldots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} \ldots + a_m z^{-m} \ldots + a_M z^{-M}}$$

Second order filter:

$$H_d(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

MATLAB FUNCTIONS:

- freqz.m : computes the frequency response
- impz.m : computes the impulse response

Example Matlab Code on Website:
dftf.m

Summary

We have learned about:

- The concept of discretisation
- The basics principles of finite difference methods
- Stability analysis for difference equations
- Analysis of numerical artefacts
- Digital impulse response and frequency response
- Relationship to filter theory, transfer functions.