Catch-up course on dynamic systems and implementation in Matlab

Lecture 1

Georgios Birpoutsoukis
In this catch-up course we will talk about...

- System, input and output
- System categorizations
- Linear vs Nonlinear
- Continuous vs Discrete
- Convolution integral
- Laplace transform
- Characteristics of transfer functions: Poles, Zeros, Order, Delay
- Common excitations: Step, Random, Impulse
- Z-transform
- Fourier transform
- Frequency Response Function
- Response of systems to sinusoidal inputs
Structure of this course

- Each lecture consists of 2 x 45 min
- First 45 min belongs to theory
- Second 45 min belongs to implementation of the theory in Matlab

- Come to the courses
- Ask questions
- Do the exercises
- Advice
What is a system? What is a signal?

\[ U \rightarrow \text{System} \rightarrow Y \]

- **U**: input/excitation to the system
- **Y**: true output/response of the system (Reaction of the system to the excitation)
- **n_u, n_y**: measurement noise (it will always be there when you measure a quantity)
- **u_m**: measured input
- **y_m**: measured output
System categorizations

- Linear VS Nonlinear
- Static VS Dynamic
- Time-invariant VS Time-varying
- Continuous VS Discrete

More categorizations can be done, e.g. lumped VS distributed, deterministic VS stochastic, e.t.c.

In this course we will mainly focus on linear time-invariant dynamic systems, both continuous and discrete.
Static vs Dynamic

- **Static system:** current output depends ONLY on current input

- **Dynamic system:** current output depends on current AND previous values of the input
  (there is memory in the system)

\[
F_{\text{spring}}(t) = k \cdot x(t)
\]

\[
F_{\text{damper}}(t) = b \cdot \dot{x}(t)
\]
Time-invariant vs Time-varying

- **Time-invariant system**: behavior between input and output does not change with time

  \[ F_{\text{spring}}(t) = k \cdot x(t) \]

  \[ F_{\text{damper}}(t) = b \cdot \dot{x}(t) \]

- **Time-varying system**: behavior between input and output changes with time

  \[ F_{\text{spring}}(t) = k(t) \cdot x(t) \]

  \[ F_{\text{damper}}(t) = b(t) \cdot \dot{x}(t) \]
Linear vs Nonlinear

Properties of linear systems:

- Homogeneity: if \( u(t) \rightarrow y(t) \) then \( \alpha u(t) \rightarrow \alpha y(t) \quad \forall \alpha \text{ constant} \)

- Superposition: if \[ \begin{align*}
  u_1(t) &\rightarrow y_1(t) \\
  u_2(t) &\rightarrow y_2(t)
\end{align*} \] then \( u_1(t) + u_2(t) \rightarrow y_1(t) + y_2(t) \)
Continuous vs Discrete

Continuous response

Response of the system available at any time instant

Discrete response

We have samples of the response only at distinct time instants

The true nature of the system should NOT be confused with HOW I CHOOSE to look at the system
Linear Continuous Time-invariant system

\[ u \rightarrow \text{System} \rightarrow y \]

The convolution integral

\[ y(t) = \int_0^\infty h(t - \tau)u(\tau)d\tau \]

Impulse response of the linear system
Laplace transform (Continuous systems)

- Not always straightforward and easy to handle the differential equations
- Transformations can be used to change an expression without losing information

\[ \mathcal{L}\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt \]

- Convolution becomes a simple product! (Very important)

\[ y(t) = \int_0^\infty h(t - \tau)u(\tau)d\tau \Rightarrow Y(s) = H(s)U(s) \]
Laplace transform (Continuous systems)

- LAPLACE TRANSFORM MAKES THE DIFFERENTIAL EQUATIONS ALGEBRAIC! 😊

\[ F(s) = \int_0^\infty f(t)e^{-st} dt \]

Laplace and derivatives

\[ \mathcal{L}\{f'(t)\} = sF(s) - f(0) \]

\[ \mathcal{L}\{f''(t)\} = s^2F(s) - sf(0) - f'(0) \]

... and in general ...

\[ \mathcal{L}\{f^{(n)}(t)\} = s^nF(s) - s^{n-1}f(0) - \cdots - f^{(n-1)}(0) \]
Laplace transform (example)

From differential to algebraic equations

\[ M\ddot{x}(t) + b\dot{x}(t) + kx(t) = F(t) \]
\[ \mathcal{L}\{M\ddot{x}(t) + b\dot{x}(t) + kx(t)\} = \mathcal{L}\{F(t)\} \]
\[ \mathcal{L}\{M\ddot{x}(t)\} + \mathcal{L}\{b\dot{x}(t)\} + \mathcal{L}\{kx(t)\} = \mathcal{L}\{F(t)\} \]

Assuming \( x(0) = \dot{x}(0) = 0 \)

\[ Ms^2X(s) + bsX(s) + kX(s) = F(s) \]
\[ (Ms^2 + bs + k)X(s) = F(s) \]

\[ \frac{X(s)}{F(s)} = \frac{1}{Ms^2 + bs + k} \]
Tables of Laplace transform

Laplace transforms of commonly used functions in the Laplace domain are available on the internet (always check the source and the content before using it)

### Laplace Transforms of Common Functions

<table>
<thead>
<tr>
<th>Name</th>
<th>( f(t) )</th>
<th>( F(s) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impulse</td>
<td>[ \begin{cases} 1 &amp; t = 0 \ 0 &amp; t &gt; 0 \end{cases} ]</td>
<td>1</td>
</tr>
<tr>
<td>Step</td>
<td>( f(t) = 1 )</td>
<td>( \frac{1}{s} )</td>
</tr>
<tr>
<td>Ramp</td>
<td>( f(t) = t )</td>
<td>( \frac{1}{s^2} )</td>
</tr>
<tr>
<td>Exponential</td>
<td>( f(t) = e^{at} )</td>
<td>( \frac{1}{s - a} )</td>
</tr>
<tr>
<td>Sine</td>
<td>( f(t) = \sin(\omega t) )</td>
<td>( \frac{1}{\omega^2 + s^2} )</td>
</tr>
</tbody>
</table>
Characteristics of transfer functions

Gain = $G(0) = \frac{1}{10}$
(check also final value theorem)

Gain = $G(0) = \frac{1}{10}$
(check also final value theorem)

$G(s) = \frac{s + 1}{s^2 + 2s + 10} e^{-\theta s}$

Zeros
$s + 1 = 0$
$s_z = -1$

Poles
$s^2 + 2s + 10 = 0$
$s_{p,1} = -1 + 3i$
$s_{p,2} = -1 - 3i$

The system has 2 poles
Order of the system?

Delay of $\theta$ seconds
Poles and System Response

\[ \text{Pole} = R(s) + I(s)i \]

- Stability/instability \( \rightarrow \) Real part of pole \( R(s) \)
- Oscillatory behavior \( \rightarrow \) Imaginary part of pole \( I(s) \)
Step response and first order system

Can we compute the values of $K$, $\tau$ and $\theta$ based on the step response?

$$G(s) = \frac{K}{\tau s + 1} e^{-\theta s}$$
Second order system

\[ G_1(s) = \frac{1}{s^2 + 5s + 6} \]

\[ G_2(s) = \frac{1}{s^2 + 0.7s + 6} \]

Different response for the same system order. Why?
Other common excitations

- Impulse
- Random
- Sinusoidal
Other common excitations

Multisine
Matlab exercises for Bruface Catch up course, 11/10/2016

These exercises are focused on the lecture of 11/10/2016 and deal with continuous time systems, step and impulse response as well as stability of a linear dynamic system (All the plots must have title, labels on the x- and y- axis, legend and grid, wherever is needed in order to make the figure clear).

- Consider the following dynamic system

\[ g(s) = \frac{s + 1}{s^2 + 0.9s + 1} \]

Questions

1) Is the system continuous or discrete?
2) What is the order of the system?
3) Which are the poles and the zeros of the system? Is the system stable or unstable? Why?
4) What is the gain of the system?
5) Verify your answers to questions 3 and 4 by exciting the system with a unit step excitation. Compute the response of the system in two ways (Verify both ways give you the same result).
6) Do you observe oscillations at the output? Why? Why not?
7) Excite the system with an impulse and verify that the system is stable (or unstable).

Have fun and do not hesitate to ask if any question/problem/comment comes up!
Catch-up course on dynamic systems and implementation in Matlab

Lecture 2

Georgios Birpoutsoukis
Today we will talk about...

- Response of systems to sinusoidal inputs
- Bode plot (Magnitude, Phase) and Frequency Response Function
- Bode plot of first, second and higher order systems
- Z-transform
Response to a sinusoidal input

In order to investigate the frequency response we need only the imaginary part of $s$

For a Linear Time Invariant system, at steady state

$$u(t) = A\sin(\omega t) \quad \Rightarrow \quad y(t) = A|G(j\omega)|\sin(\omega t + \phi)$$

$$\phi = \angle G(j\omega)$$
Bode plot (Frequency Response Function)

\[ u(t) = A \sin(\omega t) \]
\[ y(t) = A|G(j\omega)| \sin(\omega t + \phi) \]

\[
\text{Ratio of amplitudes} = R = \frac{A|G(j\omega)|}{A} = |G(j\omega)|
\]

\[
\text{Phase} = \phi = \angle G(j\omega)
\]

The ratio as well as the phase depend on the frequency of the input signal.
The response of the system is FREQUENCY-dependent.
We need a plot which depicts the ratio and phase for each frequency.
This is the Bode plot!
...and recall the superposition principle...

\[ u(t) = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t) \]

\[ y(t) = A_1 |G(j\omega_1)| \sin(\omega_1 t + \phi_1) + A_2 |G(j\omega_2)| \sin(\omega_2 t + \phi_2) \]

\[ \phi_1 = \angle G(j\omega_1) \]

\[ \phi_2 = \angle G(j\omega_2) \]
Analytic computation of the bode plot

Case of first order system

\[ G(s) = \frac{K}{\tau s + 1} \]

\[ s = j\omega \]

\[ G(j\omega) = \frac{1}{\tau \omega j + 1} = \frac{1}{1 + (\tau \omega)^2} - \frac{\tau \omega}{1 + (\tau \omega)^2}j \]

**Ratio of amplitudes**

\[ |G(j\omega)| = \frac{1}{\sqrt{1 + (\tau \omega)^2}} \]

The ratio can also be given in dB

\[ \text{Ratio in } \text{dB} = 20 \log_{10} |G(j\omega)| \]

**Phase of output signal**

\[ \phi = \angle G(j\omega) = -\arctan(\tau \omega) \]

The phase will be given in degrees
Bode plot of a first order system

\[ G(s) = \frac{K}{\tau s + 1} \]
Bode plot of a second order system

\[ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]
Effect of damping ratio

\[ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \]

\[ \omega_n = \text{const} \]

Decreasing \( \zeta \)
Effect of natural frequency

\[ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]

\[ \zeta = \text{const} \]

Increasing \( \omega_n \)
Bode plots of common transfer functions (1)

**Integrator**

\[ G(s) = \frac{1}{s} \]
Bode plots of common transfer functions (2)

**Differentiator**

\[ G(s) = s \]
Bode plots of common transfer functions (3)

**Delay**

\[ G(s) = e^{-\theta s} \]
Effect of a real zero
Effect of a real pole
...and for more complex systems...

\[ G(s) = \frac{K}{\tau s + 1} \cdot \frac{\omega_{n1}^2}{s^2 + 2\zeta_1 \omega_{n1} s + \omega_{n1}^2} \cdot \frac{\omega_{n2}^2}{s^2 + 2\zeta_2 \omega_{n2} s + \omega_{n2}^2} \]

\[
\begin{align*}
\zeta_1 &= 0.005 \\
\omega_{n1} &= 10 \\
\zeta_2 &= 0.0005 \\
\omega_{n2} &= 1000 \\
K &= 10000 \\
\tau &= 100
\end{align*}
\]
Which system is that?
Discrete time systems
Choosing the sampling frequency is crucial! An inappropriate choice can have very undesired results, even instability for an initially stable continuous system!
**Difference equations, Z transform and q-operator**

**Continuous** dynamic systems are described by **differential** equations.

\[ \ddot{y}(t) + 0.3\dot{y}(t) + 0.1y(t) = \dot{u}(t) + 2u(t) \]

**Discrete** time systems are described by **difference** equations.

\[ y[k] + 2y[k-1] + y[k-2] + 0.1y[k-3] = u[k] + 2u[k-1] + 0.1u[k-2] \]

For **continuous** time systems we apply the **Laplace transform**.

\[ \mathcal{L}\{f(t)\} = F(s) = \int_{0}^{\infty} f(t)e^{-st}dt \]

For **discrete** time systems the corresponding transform is the **Z-transform**.

\[ \mathcal{Z}\{x[k]\} = X(z) = \sum_{n=0}^{\infty} x[k]z^{-k} \]

\[ \mathcal{Z}\{x[k-n]\} = z^{-n}X(z) \]
Z-transform and the q-operator

Example

\[
\frac{Y(z)}{U(z)} = \frac{1 + 2z^{-1} + 0.1z^{-2}}{1 + 2z^{-1} + z^{-2} + 0.1z^{-3}}
\]

\[
Y(z) + 2z^{-1}Y(z) + z^{-2}Y(z) + 0.1z^{-3}Y(z) = U(z) + 2z^{-1}U(z) + 0.1z^{-2}U(z)
\]

\[
\mathcal{Z}\{y[k] + 2y[k-1] + y[k-2] + 0.1y[k-3]\} = \mathcal{Z}\{u[k] + 2u[k-1] + 0.1u[k-2]\}
\]

\[
y[k] + 2y[k-1] + y[k-2] + 0.1y[k-3] = u[k] + 2u[k-1] + 0.1u[k-2]
\]

- At some point you will find a discrete time system defined with the q operator... no worries, it is quite similar as interpretation even though it is very different mathematically

\[
\frac{y(k)}{u(k)} = \frac{1 + 2q^{-1} + 0.1q^{-2}}{1 + 2q^{-1} + q^{-2} + 0.1q^{-3}}
\]
From continuous to discrete

• There exist different ways to discretize a system
• In Matlab you have the following options: ZOH, FOH, Tustin and Impulse

**ZOH**: assumes that the signal remains constant between two sequential samples

**FOH**: first order approximation between two sequential samples

**Impulse**: the impulse response of continuous and discretized system is the same

**Tustin**: useful for control purposes, provides stable discretization
Tables of Z-transform

As for the Laplace transform, there are also tables available for the Z-transform (again check the source and the content before using the table)

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta[n]$</td>
<td>$1$</td>
</tr>
<tr>
<td>$u[n]$</td>
<td>$\frac{1}{1-z^{-1}}$</td>
</tr>
<tr>
<td>$-u[-n-1]$</td>
<td>$\frac{1}{1-z^{-1}}$</td>
</tr>
<tr>
<td>$\delta[n-m]$</td>
<td>$z^{-m}$</td>
</tr>
<tr>
<td>$a^n u[n]$</td>
<td>$\frac{1}{1-az^{-1}}$</td>
</tr>
<tr>
<td>$-a^n u[-n-1]$</td>
<td>$\frac{1}{1-az^{-1}}$</td>
</tr>
<tr>
<td>$n a^n u[n]$</td>
<td>$\frac{az^{-1}}{(1-az^{-1})^2}$</td>
</tr>
<tr>
<td>$-na^n u[-n-1]$</td>
<td>$\frac{az^{-1}}{(1-az^{-1})^2}$</td>
</tr>
</tbody>
</table>
| $a^n$             | $\begin{cases}  
                  1-a^N z^{-N} \\ 
                  1-az^{-1}  
                \end{cases}$ |
| $0 \leq n \leq N-1$, |
| $0$               | otherwise                                      |
| $\cos(\omega_0 n) u[n]$ | $\frac{1-\cos(\omega_0) z^{-1}}{1-2\cos(\omega_0) z^{-1} + z^{-2}}$ |
| $r^n \cos(\omega_0 n) u[n]$ | $\frac{1-r \cos(\omega_0) z^{-1}}{1-2r \cos(\omega_0) z^{-1} + r^2 z^{-2}}$ |
Poles of a discrete time system

<table>
<thead>
<tr>
<th>Pole-Zero Plot</th>
<th>Graph</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Pole-Zero Plot" /></td>
<td><img src="image2" alt="Graph" /></td>
<td>stable</td>
</tr>
<tr>
<td><img src="image3" alt="Pole-Zero Plot" /></td>
<td><img src="image4" alt="Graph" /></td>
<td>marginally stable</td>
</tr>
<tr>
<td><img src="image5" alt="Pole-Zero Plot" /></td>
<td><img src="image6" alt="Graph" /></td>
<td>unstable</td>
</tr>
<tr>
<td><img src="image7" alt="Pole-Zero Plot" /></td>
<td><img src="image8" alt="Graph" /></td>
<td>unstable</td>
</tr>
<tr>
<td><img src="image9" alt="Pole-Zero Plot" /></td>
<td><img src="image10" alt="Graph" /></td>
<td>unstable</td>
</tr>
<tr>
<td><img src="image11" alt="Pole-Zero Plot" /></td>
<td><img src="image12" alt="Graph" /></td>
<td>unstable</td>
</tr>
</tbody>
</table>
Why does the bode plot stop at a certain frequency?

Rules of thumb for the sampling frequency exist (typically depends on the applications):

In control applications, sampling frequency is chosen 10 times the bandwidth of the continuous time system.
Matlab exercise session for Bruface, 19/10/2016

These exercises are focused on the lectures of 11/10/2016 and 18/10/2016 and deal with continuous and discrete time systems, step and impulse response of a dynamic system, stability of a linear system as well as the construction and analysis of the Bode plot (All the plots must have title, labels on the x- and y- axis, legend and grid, wherever is needed in order to make the figure clear).

- Consider the following dynamic system

\[
g(s) = \frac{10}{s^2 + 0.1s + 10}
\]

Questions

1) Is the system continuous or discrete?
2) What is the order of the system?
3) Which are the poles and the zeros of the system? Is the system stable or unstable? Why?
4) What is the gain of the system?
5) Verify your answers to questions 3 and 4 by exciting the system with a unit step excitation. Compute the response of the system in two ways (Verify that both ways give you the same result).
6) Do you observe oscillations at the output? Why? Why not?
7) Excite the system with an impulse and verify again that the system is stable (or unstable).
8) Compute the frequency response of the system. Do you observe a resonance at the plot of the magnitude? Would you observe the same phenomenon if the system was of order 1?
9) Look at the magnitude plot of the frequency response and read from the plot the value of the norm of the transfer function at frequency 2.04 rad/sec. Excite the system with a sinusoidal input of frequency 2.04 rad/sec and amplitude 4. Check the output signal. Does it coincide with what the bode plot suggests?

- Now discretize the previous dynamic system with a sampling period of 0.05 seconds.

1) Compute the poles and the zeros of the discrete time system. Is the discrete time system stable?
2) Verify in 3 different ways that the system is stable (or unstable). (Hint! The response of a stable system does not explode)
3) Excite the system with a sinusoidal input. Choose your own amplitude and frequency. Excite the system for sufficiently long time such that you can observe both the transient and the steady-state behavior of the system. (Note that when you excite a discrete time system the step of the discrete time vector at which you excite the system should be equal to the sampling period
4) Make the bode plot of the discretized system. The plot stops at some frequency. Does this frequency coincide with what you expect?

Have fun and do not hesitate to ask if any question/problem/comment comes up!
Catch-up course on dynamic systems and implementation in Matlab

Lecture 3

Georgios Birpoutsoukis
Today we will talk about...

- Fourier Transform
- The Discrete Fourier Transform (DFT)
- DFT of common periodic and random signals
- Aliasing
- Leakage
Continuous Fourier Transform

- The spectrum, i.e. the frequency content of the signal, is most important
- We saw that the response of a system depends on the frequency contained in the input signal
- There exists a transformation which gives us access to the spectrum of a signal

\[ U(j\omega) = \int_{-\infty}^{\infty} u(t)e^{-j\omega t} \, dt \]

- \( u(t) \) : continuous analog signal
- \( \omega \) : frequency (\( \omega = 2\pi f \))

- Transforms a continuous signal from the time to the frequency domain
- The Inverse Fourier transform exists in order to go back to time
- Not really practical when working in a digital environment such as the computer
- Another transformation for the discrete time signals is necessary
Discrete Fourier Transform (DFT)

\[ U(l) = \sum_{k=0}^{N-1} u(kT_s) e^{-j2\pi \frac{kl}{N}} \]

- \( U(l) \): component of signal spectrum at frequency line \( l \), it has a magnitude and a phase
- \( T_s \): sampling period \( T_s = \frac{1}{f_s} \), \( f_s \) is the sampling frequency
- \( u(kT_s) \): signal known integer multiples of the sampling period
- \( l \): frequency line
- \( N \): number of samples of the discrete time signal
- \( S \): scaling factor (depends on the signal, see following slides)
DFT: The frequency grid

\[ U(l) = S \sum_{k=0}^{N-1} u(kT_s) e^{-j2\pi \frac{kl}{N}} \]

\[ T_s = \frac{1}{f_s} \]

N samples in time domain \[\rightarrow\] N points in frequency domain

0, \( \frac{f_s}{N} \), \( 2\frac{f_s}{N} \), \ldots, \( k\frac{f_s}{N} \), \ldots, \( f_s - \frac{f_s}{N} \)

DC component
Frequency component: \( U(1) \)
Frequency component: \( U(2) \)

This is the frequency resolution!

How can we make the grid more dense?

Why?
DFT: The scaling factor

\[ U(l) = S \sum_{k=0}^{N-1} u(kT_s)e^{-j2\pi \frac{kl}{N}} \quad T_s = \frac{1}{f_s} \]

The scaling factor is chosen based on properties of the discrete signal.

Two choices:

\[ S = \frac{1}{\sqrt{N}} \]: used for random sequences and signals where the number of frequency components grows proportionally to \( N \)

(energy of the signal is preserved, Parseval’s theorem is applied)

\[ S = \frac{1}{N} \]: used for sequences where the number of frequency components does not depend on \( N \), e.g. periodic signals

(you can recover the exact Fourier coefficients)
DFT of common signals: Sine

Blue: \( u(t) = \sin(2\pi ft) \quad f = 2\,Hz \)

Red: \( u(k) = \sin(2\pi fkT_s) \quad f_s = 10\,Hz \)

Complex conjugate of first half

\[ U(l) = \overline{U(N - l)} \]
DFT of common signals: Cosine

**Blue:** \( u(t) = \cos(2\pi ft) \quad f = 2Hz \)

**Red:** \( u(k) = \cos(2\pi fkT_s) \quad f_s = 10Hz \)

Complex conjugate of first half

\[
U(l) = \overline{U(N - l)}
\]
DFT of common signals: Multisine(1)

Continuous multisine

\[ u(t) = A_0 + \sum_{k=1}^{F} A_k \cos(2\pi k f_0 t + \phi_r) \]

- DC offset
- Careful when you excite a nonlinear system. (Why?)
- Amplitude of each cosine
- Multiples of the fundamental frequency
- Phase of each cosine

Discrete multisine

\[ u(n) = A_0 + \sum_{k=1}^{F} A_k \cos(2\pi k f_0 n T_s + \phi_r) \]

\[ = A_0 + \sum_{k=1}^{F} A_k \cos(2\pi k \frac{f_0}{f_s} n + \phi_r) \]
DFT of common signals: Multisine(2)

We can excite the desired frequency band while not violating the amplitude constraints in the time domain. Which choice for the phase would be appropriate for that purpose?

\[ u(n) = A_0 + \sum_{k=1}^{F} A_k \cos(2\pi k f_0 n T_s + \phi_r) \]

3 multisines with same amplitudes for each cosine
The alias effect in time domain

After sampling the two continuous signals, the sampled versions are exactly the same! Starting from the discrete signal, which of the two continuous signals can be reconstructed? What went wrong with the other one?
Alias in frequency domain

The sampling frequency should be minimum 2 times the value of the highest frequency contained in the continuous signal (or the Nyquist frequency must be larger than the largest frequency contained in the continuous signal).

Nyquist theorem:
The sampling frequency should be minimum 2 times the value of the highest frequency contained in the continuous signal (or the Nyquist frequency must be larger than the largest frequency contained in the continuous signal).
The leakage effect

Power leaks to neighboring frequencies because the frequency of the continuous signal is not included in the frequency grid. Remember that the grid is determined by two things: Sampling frequency \( f_s \) and Number of measurements \( N \).
The leakage effect (2)

Number of samples $N$

| Blue: $u(t) = \sin(2\pi ft)$ |
| Red: $u(k) = \sin(2\pi fkT_s)$ |

| $N = 12$ |
| $N = 13$ |

Power is still leaking and the frequency of the continuous signal is not captured by the DFT.
Finally, the true frequency of the continuous signal is again captured by the DFT. This will happen when we measure integer multiples of the period of the sinusoid. (Prove it)
We measure 1 period of cosine.
The frequency of the cosine corresponds to the 1st frequency line.
We measure now 2 periods of cosine.
The frequency of the cosine corresponds now to the 2nd frequency line.
The frequency line changes but, of course, the frequency of the continuous signal does not.
Matlab exercise session 2 for Bruface, 8/11/2016

These exercises are focused on the lecture of 26/10/2016 and deal with the implementation in Matlab of the Discrete Fourier Transform for a discrete time signal. (All the plots must have title, labels on the x- and y-axis, legend and grid, wherever is needed in order to make the figure clear).

Part 1 – Sampling frequency, number of samples and measurement time

Questions

1) We compute the DFT of a discrete time signal. How do we define the frequency resolution? What is the frequency grid that we can access with the DFT?

2) We sample a continuous multisine signal with a sampling period of 0.01 seconds/sample and we measure for 1 minute. What will be the frequency resolution of the DFT of this signal?

3) How long should I measure a continuous cosine if the sensor has a sampling frequency of 1000 Hz and I want to obtain a DFT with a resolution of 1 Hz? (Provide the answer in seconds and in samples). Shall we observe leakage in the amplitude spectrum of the DFT if the period of the continuous cosine is $T = 0.25$ sec?

Part 2 - From time to frequency domain

Questions

1) Consider two continuous sinusoids with frequencies $f_1 = 2$ Hz and $f_2 = 8$ Hz, unit amplitudes and phases $\pi$ and 0, respectively. Sample both signals with a sampling frequency $f_s = 10$ Hz in the time interval [0, 1]. Make two time domain figures with the continuous time and the discrete time signals of each sinusoid. Plot also in one figure the two discrete time signals together. What do you observe for the sampled versions of the two continuous signals? Explain your observation with the use of the DFT of the discrete time signals.

2) Consider a continuous sine wave with frequency $f_c = 2$ Hz. Sample the signal with a sampling frequency of $f_s = 10$ Hz in $N$ points with $N = 13$ and 15. Calculate for each value of $N$ the DFT of the signal and draw the amplitude spectrum with the x-axis scaled in Hz. Add to this plot also the amplitude spectrum of the original sinewave. For which value(s) of $N$ do you observe leakage? Why?

3) Consider a continuous time sinusoid of amplitude 1 and zero phase. Sample the signal with a sampling frequency $f_s = 1000$ Hz. Choose the frequency of the continuous signal such that 1 period of the signal corresponds to $N = 16$ samples. Create two discrete time signals, the first one should contain 1 period of the continuous signal and the second one should contain 2 periods of the continuous signal. Make a figure with the following 6 subplots (2,3,k):

   I. (k=1) x-axis: time (seconds), y-axis: The continuous signal together with the first discrete time signal
   II. (k=2) x-axis: frequency (Hz), y-axis: Amplitude spectrum of the DFT of the first discrete time signal
   III. (k=3) x-axis: DFT line number, y-axis: Amplitude spectrum of the DFT of the first discrete time signal
IV. (k = 4) x-axis: time (seconds), y-axis: The continuous signal together with the second discrete time signal

V. (k = 5) x-axis: frequency (Hz), y-axis: Amplitude spectrum of the DFT of the second discrete time signal

VI. (k = 6) x-axis: DFT line number, y-axis: Amplitude spectrum of the DFT of the second discrete time signal

Part 3 - From frequency to time domain

Questions

1) In this exercise we will construct the discrete time signal $u(k)$, $k = 0, 1, \ldots, N - 1$ starting from the frequency domain expression of the signal $U(k)$, $k = 0, 1, \ldots, N - 1$, using the command $\text{ifft}(U)$ (Inverse DFT - $u(k) = \sum_{l=0}^{N-1} U(l)e^{\frac{2\pi i l k}{N}}$). The goal is to construct one period of a discrete time sinusoid which has occurred after sampling a continuous time sinusoid of frequency $f_c = 62.5 \text{ Hz}$ with a sampling frequency $f_s = 1000 \text{ Hz}$. In order to do so, we go through the following steps:

- Define the number of measurements $N$ which correspond to one period of the signal.
- Construct the spectrum of the discrete time signal in the frequency domain. Remember first that the DFT of the signal will be the same size as the length of the measurements (if we measure $N_{\text{per}}$ periods then the length of the DFT will be $N \times N_{\text{per}}$). Second, remember that when we measure one period then the frequency of the continuous signal we measure will correspond to the first DFT line (not the zero-th DFT line), if we measure two periods the same frequency will correspond to the second DFT line and so on so forth. Finally, the DFT of a sampled sinusoid with amplitude $A$ and zero phase, scaled with the number of samples, is equal to $A e^{-i\frac{\pi}{2}}$ for the positive frequency and $A e^{i\frac{\pi}{2}}$ for the negative frequency (you can prove it by applying directly the formula of the inverse DFT).
- Since we construct the spectrum by using directly the scaled-with-N $\text{fft}$ coefficients, then the discrete signal obtained with $\text{ifft}(U)$ should be multiplied with the number of samples.
- Now consider again the spectrum of the discrete time signal but fill in only half the spectrum. In our case for the sinusoid, we fill in the $f\text{ft}$ coefficient corresponding to the positive frequency but we set everything else to zero. Compute now the discrete signal with the formula $2 \times \text{real}(\text{length}(U) \ast \text{ifft}(U))$ and observe that we obtain again the same discrete signal as before. This formula is related to properties of the complex numbers ($xy + \bar{x}\bar{y} = 2 \times \text{real}(xy)$). You now know two ways to go from frequency to time domain.
- Re-run your code by setting $N_{\text{per}} = 2$ or higher and observe the construction of more periods in the discrete time domain.
2) In this exercise we will construct a multisine signal starting from the frequency domain. The goal is to construct the discrete signal

\[ u(n) = \sum_{k=1}^{F} A_k \cos(2\pi k f_0 n T_s + \varphi_k) \]

with \( A_k = 1, \) \( k = 0, 1, \ldots, F, \) and a period of \( N = 1000 \) samples when measured with a sampling frequency \( f_s = 1000 \) Hz (by period of the multisine we refer to the period of the slowest cosine in the signal). The multisine should excite the frequency band \([0,0.1 f_s]\) (now you can compute \( F \) in the formula above). Choose the phase for each term in the multisine to be a random number between \([0,2\pi]\). Fill in only half the spectrum of the multisine and obtain the discrete time signal with the formula \( 2 * \text{real}(\text{length}(U) * \text{ifft}(U)) \).

Observe that the amplitude of the spectrum in the frequency domain is flat and also that the discrete signal we constructed looks more random than periodic. Finally, repeat the discrete time multisine signal twice in the time domain, so two periods of the whole discrete signal. Go back to frequency with the \( \text{fft} \) command (do not forget to define the new frequency grid!). What do you observe now for the spectrum of the signal repeated twice in time? Explain your observations by making use of the formula of the multisine.

Have fun and do not hesitate to ask if any question/problem/comment comes up!
These exercises are focused on the use of several commonly used blocks in Simulink.

1) Open a new model in Simulink and define the following continuous time system:

\[ g(s) = \frac{s + 1}{s^2 + 0.9s + 1} \]

Excite the system with a step excitation (use the block “Step”) and observe the response of the system to the step input (Define the system in two ways, both with the block “Transfer Fcn” and with the block “Zero-Pole”). Use text in the simulink model to describe the signals.

2) Repeat the same but now include also a delay at the system response of 4 sec.

3) Define the above continuous time system in Matlab. Discretize the system with a sampling frequency of 1 kHz. Obtain the numerator and denominator of the discrete time system with “tfdata”. These polynomials should be automatically loaded to Simulink in the block “Discrete Transfer Fcn”. First run the Matlab script and then run the Simulink model to compute the step response of the discrete time system. Uncheck also the “Limit data points to last” in the “History” of the “Scope”, in order to observe the whole response.

4) At this point you should have three different systems in your Simulink model. Create a subsystem (find the “Subsystem” block in the library) with three inputs (the three step signals) and three outputs (the response of each system to a step excitation). Plot the systems’ responses outside the “Subsystem” block.

5) Consider the response of the system from ex.1 and the response of the system from ex.3. Now we will plot both signals in one “Scope” block (similar to subplot(2,1,k) in Matlab). Go to the properties of the Scope and set the number of axis to 2. Connect the desired signals to the input ports and run.

6) Consider again the same two responses as in ex.5. Now we will plot the two responses in one Scope, one on top of the other. (Hint: use the block “Mux” from the library). Plot once a signal with delay and one without delay to observe the difference.

7) Use the block “Demux” at the output of the block “Mux” to get back the two signals which were introduced in the multiplexer (“Mux”). Go to “Display”,
“Signals & Ports” and check the option “Signal dimensions”. In this way you know how many different signals are included in one line in the model.

8) Use the blocks “Bus creator” and “Bus selector” to group two or more signals. What is the difference between the “Bus creator” and “Mux”? (Visit this site for more info on the difference between Bus signals and Mux signals. https://nl.mathworks.com/matlabcentral/answers/97787-what-is-the-difference-between-the-mux-demux-and-bus-creator-bus-selector-blocks-in-simulink)

9) Consider the continuous and the discrete time system defined so far. Obtain the polynomials of both transfer functions with the command “tfdata”. Open now a new model in Simulink. The polynomials of the two systems defined in Matlab should be automatically loaded to the new model in the corresponding blocks.

- We will excite the discrete time system with a random excitation. Define the random excitation u_rand in the script in Matlab. Given the sampling frequency of 1 kHz, the size of the signal u_rand should be such that the system will be excited for 15 sec. The random input should also be loaded automatically to the Simulink model with the block “From workspace”.

- Excite the continuous time system in the Simulink model with a sinusoid of amplitude 2 and frequency 10 rad/s.

- For the continuous time system use one “Scope” to observe both the input and the output (NOT on top of each other). For the discrete time system the response should be transferred back to Matlab using the block “To Workspace”. Excite also the discrete time system in Matlab with the command “filter” and compare the response to the one obtained from Simulink.

In this exercise the Simulink model should be simulated with the command “Sim” and NOT using the “Run” button in the model.
10) Finally create a negative feedback for the discrete time system using the block “Sum”. The output of the discrete time system to the input u_rand should be fed back and subtracted from the input signal (that is how you can create feedback loops when you want to e.g. control a dynamic system). Amplify the error using the block “Gain” and observe the response of the system while you change the value of the gain. Start from a gain value of 1, increase it to observe that the error between the system output and the reference signal decreases and increase it more to observe the system becoming unstable.