Frequency Domain Modeling

**Defn.** The frequency response of a linear system is the relationship between the gain and phase of a sinusoidal input and the corresponding steady state (sinusoidal) output.

\[ u = A \sin(\omega t) \]
\[ y = B \sin(\omega t + \phi) \]

**Bode plot (1940; Henrik Bode)**
- Plot gain and phase vs input frequency
- Gain is plotting using log-log plot
- Phase is plotting with log-linear plot
- Can read off the system response to a sinusoid – in the lab or in simulations
- Linearity ⇒ can construct response to any input (via Fourier decomposition)
- Key idea: do all computations in terms of gain and phase (frequency domain)
Transmission of Exponential Signals

Exponential signal: \( e^{st} = e^{(\sigma + i\omega)t} = e^{\sigma t}e^{i\omega t} = e^{\sigma t}(\cos \omega t + i \sin \omega t) \)

- Construct constant inputs + sines/cosines by linear combinations
  - Constant: \( u(t) = c = ce^{0t} \)
  - Sinusoid: \( u(t) = A \sin(\omega t) = \frac{A}{2i}(e^{i\omega t} - e^{-i\omega t}) \)
  - Decaying sinusoid: \( u(t) = Ae^{-\sigma t} \sin(\omega t) \)

- Exponential response can be computed via the convolution equation

\[
x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Be^{\sigma \tau}d\tau
\]

\[
= e^{At}x(0) + e^{At}(sI - A)^{-1}e^{(sI-A)\tau}|_{\tau=0}B
\]

\[
= e^{At}x(0) + e^{At}(sI - A)^{-1}\left(e^{(sI-A)\tau} - I\right)B
\]

\[
= e^{At}\left(x(0) - (sI - A)^{-1}B\right) + (sI - A)^{-1}Be^{st}
\]

\[
y(t) = Cx(t) + Du(t)
\]

\[
= Ce^{At}\left(x(0) - (sI - A)^{-1}B\right) + \left(C(sI - A)^{-1}B + D\right)e^{st}
\]

Transfer Function and Frequency Response

Exponential response of a linear state space system

\[
y(t) = Ce^{At}\left(x(0) - (sI - A)^{-1}B\right) + \left(C(sI - A)^{-1}B + D\right)e^{st}
\]

- Steady state response is proportional to exponential input \( \Rightarrow \) look at input/output ratio
  - \( G(s) = C(sI - A)^{-1}B + D \) is the transfer function between input and output

Frequency response

\[
u(t) = A \sin \omega t = \frac{A}{2i}(e^{i\omega t} - e^{-i\omega t})
\]

\[
y_{ss}(t) = \frac{A}{2i}\left(G(i\omega)e^{i\omega t} - G(-i\omega)e^{-i\omega t}\right)
\]

\[
= A \cdot |G(i\omega)|\sin(\omega t + \arg G(i\omega))
\]

Common transfer functions

\[
\begin{align*}
\dot{y} &= u \\
y &= \dot{u} \\
\dot{y} + ay &= u \\
\dot{y} &= u \\
\dot{y} + 2\zeta \omega_n \dot{y} + \omega_n^2 y &= u \\
y &= k_p u + k_d \dot{u} + k_i \int u \\
y(t) &= u(t - \tau) + e^{-\tau s}
\end{align*}
\]
Transfer Function Properties

**Thm.** The transfer function for a linear system \( \Sigma=(A,B,C,D) \) is given by

\[
G(s) = C(sI - A)^{-1}B + D \quad s \in \mathbb{C}
\]

**Thm.** The transfer function \( G(s) \) corresponding to \( \Sigma=(A,B,C,D) \) has the following properties:

- \( H(s) \) is a ratio of polynomials \( n(s)/d(s) \) where \( d(s) \) is the characteristic equation for the matrix \( A \) and \( n(s) \) has order less than or equal to \( d(s) \).
- The steady state frequency response of \( \Sigma \) has gain \(|G(i\omega)|\) and phase \( \arg G(i\omega) \):
  
  \[
  u = A\sin(\omega t) \\
  y = |G(i\omega)|A\sin(\omega t + \arg G(i\omega)) + \text{transients}
  \]

**Remarks**

- Formally, can show that \( G(s) \) is the Laplace transform of the impulse response of \( \Sigma \)
- Often write “\( y = G(s)u \)” for \( Y(s) = G(s)U(s) \), where \( Y(s) \) & \( U(s) \) are Laplace transforms of \( y(t) \) and \( u(t) \). (Multiplication in Laplace domain corresponds to convolution.)
- MATLAB: \( G = \text{ss2tf}(A, B, C, D) \)

Series Interconnections

**Q:** what happens when we connect two systems together in series?

\[
\begin{align*}
\dot{x}_1 &= A_1 x_1 + B_1 u_1 \\
\dot{y}_1 &= C_1 x_1 + D_1 u_1
\end{align*}
\]

\[
\begin{align*}
\dot{x}_2 &= A_2 x_2 + B_2 u_2 \\
\dot{y}_2 &= C_2 x_2 + D_2 u_2
\end{align*}
\]

\[
\begin{align*}
\text{u} &= A\sin(\omega t) \\
\text{y} &= g_1 A\sin(\omega t + \phi_1)
\end{align*}
\]

\[
\begin{align*}
\text{y}_2 &= g_1 g_2 A x \\
&= g_1 g_2 A \sin(\omega t + \phi_1 + \phi_2)
\end{align*}
\]

**A:** Transfer functions multiply

- Gains multiply, phases add
- Generally: transfer functions well formulated for frequency domain interconnections
  - Convolution \( \rightarrow \) multiplication
- MATLAB/python: \( G = \text{series}(G_1, G_2) \)
Feedback Interconnection

\[ \dot{x} = u = r - ay = -ax + r \]
\[ y = x \]

Frequency response
\[ r = A \sin(\omega t) \]
\[ y = \left| \frac{1}{\sqrt{a^2 + \omega^2}} \right| \sin \left( \omega t - \tan^{-1} \left( \frac{\omega}{a} \right) \right) \]

- MATLAB: \( G = \text{feedback(sys, a)} \) ← works for either state space or transfer functions

Poles and Zeros

\[ \dot{x} = Ax + Bu \]
\[ y = Cx + Du \]
\[ G(s) = \frac{n(s)}{d(s)} \]
\[ d(s) = \det(sI - A) \]

- Roots of \( d(s) \) are called poles of \( G(s) \)
- Roots of \( n(s) \) are called zeros of \( G(s) \)

Poles of \( G(s) \) determine the stability of the (closed loop) system
- Denominator of transfer function = characteristic polynomial of state space system
- Provides easy method for computing stability of systems
- Right half plane (RHP) poles (\( \text{Re} > 0 \)) correspond to unstable systems

Zeros of \( G(s) \) related to frequency ranges with limited transmission
- A pure imaginary zero at \( s = i\omega \) blocks any output at that frequency (\( G(i\omega) = 0 \))
- Zeros provide limits on performance, especially RHP zeros (more on this later)

MATLAB: pole(G), zero (G), pzmap(G)
Sketching the Bode Plot for a Transfer Function (1/2)

Evaluate transfer function on imaginary axis

\[ M = |G(i\omega)|, \quad \phi = \arctan \frac{\text{Im } G(i\omega)}{\text{Re } G(i\omega)} \]

- Plot gain (M) on log/log scale
- Plot phase (\(\phi\)) on log/linear scale
- Piecewise linear approximations available

\[ G(s) = \frac{1}{s^k} \]

Sketching the Bode Plot for a Transfer Function (2/2)

Complex poles

\[ G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \]

Ratios of products

\[ G(s) = \frac{b_1(s) b_2(s)}{a_1(s) a_2(s)} \]

\[ \log |G(i\omega)| \approx \begin{cases} 0 & \text{if } \omega \ll \omega_0, \\ 2 \log \omega_0 - 2 \log \omega & \text{if } \omega \gg \omega_0, \\ -180 & \text{if } \omega \gg \omega_0. \end{cases} \]

\[ \angle G(i\omega) \approx \begin{cases} 0 & \text{if } \omega \ll \omega_0, \\ -90 & \text{if } \omega \gg \omega_0. \end{cases} \]

\[ G(s) = \frac{k(s + b)}{(s + a)(s^2 + 2\zeta\omega_0 s + \omega_0^2)} \quad a \ll b \ll \omega_0. \]
Example: Coupled Masses

\[ H_{q_1f} = \frac{0.04}{s^4 + 0.08s^3 + 0.8016s^2 + 0.032s + 0.12} \]
\[ H_{q_2f} = \frac{0.2s^2 + 0.008s + 0.08}{s^4 + 0.08s^3 + 0.8016s^2 + 0.032s + 0.12} \]

Poles (Hq1f and Hq2f)
- \(-0.0200 \pm 0.7743j\)
- \(-0.0200 \pm 0.4468j\)

Zeros (Hq2f)
- \(-0.0200 \pm 0.6321j\)

Interpretation
- Zeros in Hq2f give low response at \( \omega \approx 0.6321 \)

MATLAB: `bode(H)`

Block Diagram Algebra

Basic idea: treat transfer functions as multiplication, write down equations

\[
\begin{align*}
    &\text{d} \\
    &\text{r} \\
    &C(s) \quad \text{u} \quad P(s) \quad y \\
    &\text{e} \\
    &\text{u} = d + C(s)e \\
    &y = P(s)u \\
    &e = r - y
\end{align*}
\]

Manipulate equations to compute desired signals

\[
\begin{align*}
    e &= r - y \\
    &= r - P(s)u \\
    &= r - P(s)(d + C(s)e)
\end{align*}
\]

\[
\begin{align*}
    (1 + P(s)C(s))e &= r - P(s)d \\
    e &= \frac{1}{1 + P(s)C(s)}(r - \frac{P(s)}{1 + P(s)C(s)}d)
\end{align*}
\]

Note: linearity gives superposition of terms

Algebra works because we are working in frequency domain
- Time domain (ODE) representations are not as easy to work with
- Formally, all of this works because of Laplace transforms (see text)
### Block Diagram Algebra

<table>
<thead>
<tr>
<th>Type</th>
<th>Diagram</th>
<th>Transfer function</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Series</strong></td>
<td><img src="image" alt="Series Diagram" /></td>
<td>(H_{y_2u_1} = H_{y_2y_1} H_{y_1u_1} = \frac{n_1 n_2}{d_1 d_2})</td>
</tr>
<tr>
<td><strong>Parallel</strong></td>
<td><img src="image" alt="Parallel Diagram" /></td>
<td>(H_{y_3y_1} = H_{y_3y_1} + H_{y_3u_1} = \frac{n_1 d_2 + n_2 d_1}{d_1 d_2})</td>
</tr>
<tr>
<td><strong>Feedback</strong></td>
<td><img src="image" alt="Feedback Diagram" /></td>
<td>(H_{y_3r} = \frac{H_{y_3y_1}}{1 + H_{y_3y_1} H_{y_3u_1}} = \frac{n_1 d_2}{n_1 n_2 + d_1 d_2})</td>
</tr>
</tbody>
</table>

- These are the basic manipulations needed; some others are possible
- Formally, could work all of this out using the original ODEs (⇒ nothing really new)

### MATLAB/Python manipulation of transfer functions

**Creating transfer functions**
- \([\text{num}, \text{den}] = \text{ss2tf}(A, B, C, D)\)
- \(\text{sys} = \text{tf}(\text{num}, \text{den})\)
- \(\text{num}, \text{den} = [1 \ a \ b] \rightarrow s^2 + as + b\)

**Interconnecting blocks**
- \(\text{sys} = \text{series}(\text{sys1}, \text{sys2}), \text{parallel}, \text{feedback}\)

**Computing poles and zeros**
- \(\text{pole(sys)}, \text{zero(sys)}\)
- \(\text{pzmap(sys)}\)

**I/O response**
- \(\text{initial(sys)}\)
- \(\text{step(sys)}\)
- \(\text{lsim(sys)}\)
- \(\text{bode(sys)}\)
Control Analysis and Design Using Transfer Functions

Transfer functions provide a method for “block diagram algebra”
- Easy to compute transfer functions between various inputs and outputs
  - Her(s) is the transfer function between the reference and the error
  - Hed(s) is the transfer function between the disturbance and the error

Transfer functions provide a method for performance specification
- Since transfer functions provide frequency response directly, it is convenient to work in the “frequency domain”
  - Her(s) should be small in the frequency range 0 to 10 Hz (good tracking)

Key idea: perform all analysis and design for linear systems in “frequency domain”
- Convert specifications on time response to specifications on frequency response
- “Shape” the frequency response by design of controller transfer function (Ch 10-13)

Summary: Frequency Response & Transfer Functions

\[
\dot{x} = Ax + Bu \\
y = Cx + Du \\
x(0) = 0
\]

\[
y_{ss} = A \cdot |G(i\omega)| \times \\
\sin(\omega t + \arg G(i\omega))
\]

\[
G(s) = C(sI - A)^{-1}B + D \\
G_{y2u1} = G_{y2u2}G_{y1u1} = \frac{n_1n_2}{d_1d_2}
\]