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.

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

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

- Exponential response can be computed via the convolution equation

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

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

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

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

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

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

Transfer Function and Frequency Response

Exponential response of a linear state space system

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

\(\text{transient}\phantom{=}\text{steady state}\)

Transfer function

- 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

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

\[y_{ss}(t) = \frac{A}{2i} \left|G(i\omega)\right| e^{i\omega t} = A \cdot \left|G(i\omega)\right| \sin(\omega t + \arg G(i\omega))\]

<table>
<thead>
<tr>
<th>Common transfer functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\dot{y} = u)</td>
</tr>
<tr>
<td>(y = \dot{u})</td>
</tr>
<tr>
<td>(\dot{y} + ay = u)</td>
</tr>
<tr>
<td>(\dot{y} = u)</td>
</tr>
<tr>
<td>(\ddot{y} + 2\xi\omega_n\dot{y} + \omega_n^2y = u)</td>
</tr>
<tr>
<td>(y = kp + k_d\dot{u} + k_i \int u)</td>
</tr>
<tr>
<td>(y(t) = u(t - \tau))</td>
</tr>
</tbody>
</table>
Circuit dynamics (Kirchoff’s laws):

\[
\frac{v_1 - v}{R_1} = \frac{v - v_2}{R_2} \quad \text{and} \quad v_2 = G(s)v
\]

\[
\frac{v_2}{v_1} = \frac{R_2G(s)}{R_1 + R_2 + R_1G(s)} = \frac{R_2ak}{R_1 + (R_1 + R_2)(s + a)}.
\]

- Algebraic manipulation can be used as long as we assume exponential signals and all of the components (blocks) are linear
- Transfer function between input and output shows gain-bandwidth tradeoff
- Homework: derive transfer function for a PI controller using an op amp

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 \( \text{arg} \ G(i\omega) \):
  
  \[
  u = A\sin(\omega t) \\
  y = |G(i\omega)|A\sin(\omega t + \text{arg} G(i\omega)) + \text{transients}
  \]

**Remarks**

- Formally, can show that \( G(s) \) is the Laplace transform of the impulse response of \( \Sigma \)
- "\( y = G(s)u \)" is formally \( Y(s) = G(s)U(s) \) where \( Y(s) \) and \( U(s) \) are the Laplace transforms of \( y(t) \) and \( u(t) \). (Multiplication in the Laplace domain corresponds to convolution.)
Series Interconnections

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

\[ \begin{align*}
  x_1 &= A_1 x_1 + B_1 u_1 \\
  y_1 &= C_1 x_1 + D_1 u_1 \\
  x_2 &= A_2 x_2 + B_2 u_2 \\
  y_2 &= C_2 x_2 + D_2 u_2
\end{align*} \]

A: Transfer functions multiply
- Gains multiply
- Phases add
- Generally: transfer functions well formulated for frequency domain interconnections

Feedback Interconnection

State space derivation
\[ \begin{align*}
  \dot{x} &= u = r - ay = -ax + r \\
  y &= x
\end{align*} \]

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

Transfer function derivation
\[ \begin{align*}
  y &= \frac{u}{s} = \frac{r - ay}{s} = \frac{r}{s} = G(s) r
\end{align*} \]

Frequency response
\[ y = |G(i\omega)| \sin(\omega t + \angle G(i\omega)) \]
Poles and Zeros

\[ \dot{x} = Ax + Bu \quad H(s) = \frac{n(s)}{d(s)} \quad \text{Roots of } d(s) \text{ are called poles of } H(s) \]

\[ y = Cx + Du \quad d(s) = \det(sI - A) \quad \text{Roots of } n(s) \text{ are called zeros of } H(s) \]

**Poles of** \( H(s) \) **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** \( H(s) \) **related to frequency ranges with limited transmission**

- A pure imaginary zero at \( s = j\omega \) blocks any output at that frequency \((G(j\omega) = 0)\)
- Zeros provide limits on performance, especially RHP zeros (more on this later)

\[ H(s) = k \frac{s^2 + b_3 s + b_2}{s^4 + a_4 s^3 + a_3 s^2 + a_2 s + a_1} \]

**Example: Coupled Masses**

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

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

**Poles** \( (H_{q_1f} \text{ and } H_{q_2f}) \)

- \(-0.0200 \pm 0.7743j\)
- \(-0.0200 \pm 0.4468j\)

**Zeros** \( (H_{q_1f}) \)

- \(-0.0200 \pm 0.6321j\)

**Interpretation**

- Zeros in \( H_{q_2f} \) give low response at \( \omega \approx 0.6321 \)
Transfer functions provide a method for “block diagram algebra”
- Easy to compute transfer functions between various inputs and outputs
  - $H_{er}(s)$ is the transfer function between the reference and the error
  - $H_{ed}(s)$ 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”
  - $H_{er}(s)$ should be small in the frequency range 0 to 10 Hz (good tracking)

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

Manipulate equations to compute desired signals
\[
\begin{align*}
\text{Note: linearity} & \quad \text{gives superposition of terms} \\
\end{align*}
\]

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 book)
### Block Diagram Algebra

<table>
<thead>
<tr>
<th>Type</th>
<th>Diagram</th>
<th>Transfer function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Series</td>
<td><img src="image" alt="Series Diagram" /></td>
<td>$H_{y_2u_1} = H_{y_2u_1}H_{y_1u_1} = \frac{n_1n_2}{d_1d_2}$</td>
</tr>
<tr>
<td>Parallel</td>
<td><img src="image" alt="Parallel Diagram" /></td>
<td>$H_{y_2u_1} = H_{y_2u_1} + H_{y_2u_1} = \frac{n_1d_2 + n_2d_1}{d_1d_2}$</td>
</tr>
<tr>
<td>Feedback</td>
<td><img src="image" alt="Feedback Diagram" /></td>
<td>$H_{y_2u_1} = \frac{H_{y_2u_1}}{1 + H_{y_2u_1}H_{y_2u_1}} = \frac{n_1d_2}{n_1n_2 + d_1d_2}$</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 manipulation of transfer functions

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

**Interconnecting blocks**
- `sys= series(sys1, sys2), parallel, feedback`

**Computing poles and zeros**
- `pole(sys), zero(sys)`
- `pzmap(sys)`

**I/O response**
- `step(sys), bode(sys)`

```latex
\begin{align*}
\text{tf}(\text{sys}) & \quad \text{Transfer function:} \\
& \quad \frac{1}{s^2 + 0.2 s + 1}
\end{align*}
```
Example: Engine Control of a GM Astro

\[ H_{0,T_e}(s) = \frac{Kr}{J_c J_s s^3 + J_c B_1 s^2 + (J_c K + K J_i) s + K B_1} \]

Summary: Frequency Response & Transfer Functions

\[ u = A \sin(\omega t) \]

\[ \begin{align*}
\dot{x} &= Ax + Bu \\
y &= Cx + Du \\
x(0) &= 0
\end{align*} \]

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

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

\[ G_{y_2u_1} = G_{y_2u_2} G_{y_1u_1} = \frac{n_1 n_2}{d_1 d_2} \]