INTRODUCTION TO ROBUST CONTROL

7.1: Review of SISO Nyquist stability

- The control designs we have investigated are “optimal”.
- Don’t let that give you a false sense of security!
- Are they robust? Do they work well even if our plant model is inaccurate, or if it changes slightly over time?
- We will first review Nyquist, look at stability margins for LQR and LQG and look for ways to improve them.
- Nyquist stability criterion is a graphical way to determine stability of a feedback system.
- Originally developed as a method to determine stability without having to solve for closed-loop poles (difficult in 1932!).
- Nyquist still useful for several reasons:
  - Gain margin and phase margin—measures of stability and robustness—readily determined from plot.
  - Can be applied to systems with time delays (Routh cannot).
  - Modifications to controller frequency response which improve gain- and phase-margin may be readily observed from Nyquist map.
- We primarily use Nyquist to observe $GM$ and $PM$. Indicators of robustness.
The Nyquist test

- First, compute the loop transfer function \( L(s) = G(s)K(s) \).
- Plot the loop frequency response \( L(j\omega) = L(s)|_{s=j\omega} \) as a polar plot.
  That is, plot the set of points \((\Re\{L(j\omega)\}, \Im\{L(j\omega)\})\) for \(-\infty < \omega < \infty\).
- Resulting plot is a “Nyquist plot,” or “Nyquist map.”
- \( N_p \) is the number of open-loop unstable poles in \( L(s) \).
- \( N_c \) is the number of clockwise encirclements of \(-1\) in the Nyquist map.
- \( N_z = N_p + N_c \) is the number of closed-loop unstable poles. System stable iff \( N_z = 0 \).
- Special cases when \( L(s) \) has poles on the \( j\omega \)-axis. Won’t worry about them here.

**EXAMPLE:** \( K(s) = 1 \).

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

or, \( L(j\omega) = \frac{5}{(j\omega + 1)^2} \)

<table>
<thead>
<tr>
<th>( \omega )</th>
<th>( \Re{L(j\omega)} )</th>
<th>( \Im{L(j\omega)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>5.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0088</td>
<td>4.9988</td>
<td>-0.0879</td>
</tr>
<tr>
<td>0.0191</td>
<td>4.9945</td>
<td>-0.1908</td>
</tr>
<tr>
<td>0.0415</td>
<td>4.9742</td>
<td>-0.4135</td>
</tr>
<tr>
<td>0.0902</td>
<td>4.8797</td>
<td>-0.8872</td>
</tr>
<tr>
<td>0.1959</td>
<td>4.4590</td>
<td>-1.8172</td>
</tr>
<tr>
<td>0.4258</td>
<td>2.9333</td>
<td>-3.0513</td>
</tr>
<tr>
<td>0.9253</td>
<td>0.2086</td>
<td>-2.6856</td>
</tr>
<tr>
<td>2.0108</td>
<td>-0.5983</td>
<td>-0.7906</td>
</tr>
<tr>
<td>4.3697</td>
<td>-0.2241</td>
<td>-0.1082</td>
</tr>
<tr>
<td>9.4957</td>
<td>-0.0536</td>
<td>-0.0114</td>
</tr>
<tr>
<td>20.6351</td>
<td>-0.0117</td>
<td>-0.0011</td>
</tr>
<tr>
<td>500.0000</td>
<td>-0.0000</td>
<td>-0.0000</td>
</tr>
</tbody>
</table>
■ No encirclements of $-1$, $N_c = 0$.
■ No open-loop unstable poles $N_p = 0$.
■ $N_z = N_c + N_p = 0$. Closed-loop system is stable.

**Gain margin**
■ Uncertainty in plant may be modeled as
\[ G(s) = gG_o(s) \]
where $G_o(s)$ is the nominal plant, $g$ is an unknown gain.
■ Nyquist stability may be applied to the system
  - By plotting map for every admissible gain $g$.
  - By realizing that $g$ only magnifies/shrinks the map and by revising the stability rule.

**NEW RULE:**
- $N_p$ is the number of open-loop unstable poles in $L(s)$. $N_c$ is the number of clockwise encirclements of $-1/g$ and $N_z$ is number of closed-loop unstable poles. System stable iff $N_z = N_p + N_c = 0$.

**DEFINITION:** The gain margin $GM^+$ is the minimum gain $> 1$ that results in an unstable closed-loop system.

**DEFINITION:** The downside gain margin $GM^-$ is the maximum positive gain $< 1$ that results in an unstable closed-loop system.

**EXAMPLE:** Consider a plant (with two open-loop unstable poles) and controller
The gain $g$ is uncertain but bounded $0.8 \leq g \leq 1.2$.

- The Nyquist map for this system is plotted (for $g = 1$).

- The map crosses the real axis at $-0.25$ and at $-1.8$.
- $GM^+ = 1/0.25 = 4.0$.
  $GM^- = 1/1.8 = 0.56$.
- The system is therefore “robustly stable.”

**Phase margin**

- Uncertainty may also be modeled as an uncertain phase shift
  
  $G(s) = e^{-j\phi}G_o(s)$.

- This is a simplified model of an uncertain delay $G(s) = e^{-s\tau}G_o(s)$ or $G(j\omega) = e^{-j\omega\tau}G_o(j\omega)$, which is frequency dependent and harder to analyze.

- Multiplying a frequency response by $e^{-j\phi}$ results in a phase shift/rotation of $\phi$ of each point in the Nyquist map.

- System becomes unstable when rotation causes map to go through the point $-1$.

**DEFINITION:** The phase margin $PM$ is the minimum amount of phase shift added to $L(s)$ that results in an unstable closed-loop system.
Phase margin is the amount of rotation required to cause instability. Determined by intersection point between Nyquist map and unit circle.

Many Nyquist plots are like this one. Increasing loop gain magnifies the plot.

$GM = 1/(\text{distance between origin and place where Nyquist map crosses real axis})$.

If we increase gain, Nyquist map “stretches” and we may encircle $-1$. 
7.2: Robustness of LQR

- To date, we have analyzed our controllers by looking at the pole locations and time-domain performances.
- We can also look at the frequency-domain properties to get many new insights.

Define

\[ L(s) = K(sI - A)^{-1}B_u \]  
Loop transfer function

\[ C(s) = C_z(sI - A)^{-1}B_u \]  
Cost transfer function \( G_{zu}(s) \)

- \( L(s) \) is the loop we are closing.
- \( C(s) \) describes how control inputs influence the cost (state).
- The two transfer functions are related in a special way:

\[
[I + L(-s)]^T R[I + L(s)] = R + C^T(-s)C(s).
\]

This is called the KALMAN FREQUENCY-DOMAIN EQUALITY.

Proof of the Kalman frequency-domain equality

- Our main interest is to show how to relate optimal steady-state time-domain solution to frequency domain.
- Optimal steady-state control solves
\[ u = -Kx \]
\[ K = R^{-1}B_u^TP \]
\[ 0 = -A^TP - PA - Q + PB_uR^{-1}B_u^TP, \]

and we make the assumption that \( Q = C_z^TC_z \).

- Introduce Laplace variable \( \pm sP \).

\[
\underbrace{P(sI - A)}_{(1)} + \underbrace{(-sI - A^T)P - Q}_{(2)} + \underbrace{PB_uR^{-1}B_u^TP}_{(3)} = 0.
\]

- Pre-multiply by \( B_u^T(-sI - A^T)^{-1} \); post-multiply by \( (sI - A)^{-1}B_u \).

\[
(1) \quad B_u^T(-sI - A^T)^{-1}PB_u = B_u^T(-sI - A^T)^{-1}PB_uR^{-1}R = L^T(-s)R.
\]

\[
(2) \quad B_u^T(-sI - A^T)^{-1}Q(sI - A)^{-1}B_u = B_u^T(-sI - A^T)C_z^TC_z(sI - A)^{-1}B_u.
\]

Then, \( (2) = C^T(-s)C(s) \).

\[
(3) \quad B_u^T(-sI - A^T)^{-1}PB_uR^{-1}B_u^TP(sI - A)^{-1}B_u = B_u^T(-sI - A^T)^{-1}K^TRK(sI - A)^{-1}B_u = L^T(-s)RL(s)
\]

- The A.R.E. becomes

\[ RL(s) + L^T(-s)R + L(-s)^TRL(s) = C^T(-s)C(s). \]

- Complete the square

\[ [I + L(-s)]^TR[I + L(s)] = R + C^T(-s)C(s). \]

- Consider the case with only one input. \( L(s) \) is scalar. Let \( s = j\omega \).

\[ [1 + L(-j\omega)]R[1 + L(j\omega)] = R + C^T(-j\omega)C(j\omega). \]

- \( R \) is a scalar, and \( R > 0 \) so we can divide
\[
[1 + L(-j\omega)][1 + L(j\omega)] = 1 + \frac{1}{R}C^T(-j\omega)C(j\omega)
\]
\[
|1 + L(j\omega)|^2 = 1 + \frac{1}{R}|C(j\omega)|^2.
\]

- Since \(\frac{1}{R}|C(j\omega)|^2 \geq 0\) then we get the inequality
  \[
  |1 + L(j\omega)|^2 \geq 1.
  \]

- What does this mean?

- \(1 + L(j\omega)\) is the vector from the point \((-1, 0)\) to \(L(j\omega)\) in the \(s\)-plane.

- The KFDE says the \(L(j\omega)\) is excluded from the unit circle centered at \((-1, 0)\). Recall that the point \((-1, 0)\) is the critical point for stability analysis using Nyquist.

- For stability, we need to determine how many times the loop transfer function encircles the point \((-1, 0)\).

- We know that the optimum controller stabilizes the system, so counting encirclements will not be very enlightening.

- More important: What if the system changes or our model is wrong—will the number of encirclements be different than we expected? Instability.

- Basic robustness question.

- But, note that LQR guarantees that \(L(j\omega)\) is some distance from the critical point so it will take quite large changes to the system to change number of encirclements.
The optimal LQR controller has very large gain/phase margins.

- If open-loop system is stable, then any \( g \in (0, \infty) \) yields a stable closed-loop system.
- If open-loop system is unstable, then any \( g \in (1/2, \infty) \) yields a stable closed-loop system.
- LQR phase margin of at least \( \pm 60^\circ \).

**LQR Gain Margin:** Consider an example system with two unstable O.L. poles. Nyquist plot terminates at origin, stays outside the unit circle at \(-1\), and encircles the \(-1\) point twice.

- The negative real axis intercept is to the left of \(-2\) \(\Rightarrow\) We could cut the gain in half and the number of encirclements would not change.

- Increasing the gain would move the intercept further left \(\Rightarrow\) \( g \in (1/2, \infty) \) good.

**LQR Phase Margin:** The best way to look at phase margin is to look at the possible values of the loop phase when loop gain is 1.

- Anywhere on the unit circle centered at the origin outside the unit circle centered at \(-1\).
- Immediately clear that phase margin \( \geq 60^\circ \).
- Note: This does not imply that you can change the gain and phase simultaneously!
- There are similar statements for MIMO systems; not as enlightening

\[
[I + L(-s)]^T R [I + L(s)] \geq R, \quad R = \rho I
\]

\[
\sigma[I + L(j\omega)] \geq 1 \quad \forall \omega.
\]

- Singular values measure “size” of MIMO transfer-function matrix.
7.3: Robustness of LQG; LTR

- LQG offers a great way to design controllers for MIMO systems to achieve some desired PERFORMANCE specification.
- However, optimal controllers can be VERY sensitive to model errors.

**EXAMPLE:** Originally by Doyle, guru of robust control.

\[
\dot{x}(t) = Ax(t) + B_u u(t) + B_w w(t) \\
y(t) = C_y x(t) + v(t)
\]

with \(w, v\) white.

\[
A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B_u = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad B_w = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C_y = \begin{bmatrix} 1 & 0 \end{bmatrix},
\]

\[
Q = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad S_w = Q, \quad S_v = R.
\]

- Steady-state LQE/LQR gains are

\[
K = \bar{K} \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \bar{K} = \text{fn}(R)
\]

\[
L = \bar{L} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \bar{L} = \text{fn}(R).
\]

- This is fine so far, but what do the gain and phase margins look like? First look at example when \(R = 1\). (MATLAB code in appendix).
- Secondly, look at example when $R = 0.01$.

- Example shows that LQR retains guaranteed stability margin, but that stability margin of “optimal” LQG system may be arbitrarily small. (!)
- This does not mean that every LQG controller will have small margins.
- This does mean that you should be careful and check!

**Sensitivity and robustness**

- Consider the situation where we design the controller using plant dynamics $A_o$ but it turns out that the true dynamics are at $A_o + \Delta A$.
- Model uses $\dot{x} = A_o x + B_u u + B_w w$. Compensator uses $A_c = A_o - B_u K - LC_y$. 
Then, closed-loop dynamics are (by separation theorem)

\[
\begin{bmatrix}
A_0 - B_u K & B_u K \\
0 & A_0 - LC_y
\end{bmatrix}.
\]

The model tells us we are stable and doing well.

Actual system has dynamics \( \dot{x} = (A_0 + \Delta A)x + B_u u + B_w w. \)

Then, the closed-loop dynamics are (trust me!)

\[
\begin{bmatrix}
A_0 + \Delta A - B_u K & B_u K \\
\Delta A & A_0 - LC_y
\end{bmatrix}.
\]

Dynamics not decoupled! We don’t know where the eigenvalues are!

Danger, Will Robinson! Danger!

Similar with \( \Delta B_u \) and \( \Delta C_y. \)

What can we do?

- Some approximate methods to improve stability margins of LQG.
- Ways to guarantee robustness. (Performance in the face of uncertainty).
- Tough to discuss in a class or two. We will give a brief preview.

**Loop transfer recovery (LTR)**

- Method to modify LQG controller to recover robustness of LQR/ LQE.
- LTR generates suboptimal controllers with respect to LQG since any modification to original design is suboptimal.
- LTR provides a family of controllers with a range of robustness properties \( \Rightarrow \) Select controller with best compromise between robustness and performance.
Can best understand LTR by considering the following system:

- Make LQG “look like” LQR by cutting feedback path to Kalman filter, making Kalman filter “sufficiently fast” that its dynamics may be ignored.
- Can accomplish this by adding fictitious noise \( w_f(t) \) to control input \( u(t) \) during design (only!).
- This noise reduces filter’s reliance on \( u(t) \) and also makes filter faster.

\[
\dot{x}(t) = Ax(t) + Bu(t) + Bw(t) + Bw_f(t)
\]

\[
= Ax(t) + Bu(t) + \begin{bmatrix} B_w & B_u \end{bmatrix} \begin{bmatrix} w(t) \\ w_f(t) \end{bmatrix},
\]

and

\[
S \begin{bmatrix} w \\ w_f \end{bmatrix} = \begin{bmatrix} S_w & 0 \\ 0 & S_{w_f}I \end{bmatrix}.
\]

- LTR increases spectral density of fictitious noise until acceptable robustness is achieved.
- Can show (see text) that robustness approaches LQR robustness if \# measurements \( \geq \# \) control inputs and plant is minimum-phase.

**Alternate LTR**

- An alternative form of LTR uses the cost function

\[
J = \mathbb{E} \left[ x^T(\infty) \left( Q + \rho C_y^T C_y \right) x(\infty) + u^T(\infty) Ru(\infty) \right].
\]
The added term $\rho C_y^T C_y$ increases robustness but results in sub-optimal control with respect to original cost function.

Can use this version of LTR when # measurements $\leq$ # control inputs and plant is minimum-phase.

**Frequency-shaped LTR**

- The fictitious noise added to LTR may be restricted to a certain frequency band by adding filter states to the problem.
- Models frequency-dependent uncertainty in plant model (often the case).

**Other approximate methods to improve LQR robustness**

1. Look at loop transfer function to see if there is a pole/zero cancellation problem. Use perturbed models and see if compensator zero missing plant pole. Reduce compensator authority.

2. Make several models of the system

$$\dot{x}_i = A_i x_i + B_{ui} u + B_{wi} w, \quad i = 1 \ldots m.$$  

- Set up $J_i$ for each.
- Use the models to capture something about the uncertainty.
- Effective, but requires numerical search. Takes a while.

3. Guaranteed methods?

- Much research going on right now.
7.4: Modeling uncertainty

Modeling unstructured uncertainty

- GM and PM are special cases in modeling uncertainty in a plant.
- Many other types exist ⇒ Here we consider perturbations to a nominal plant model.
- A perturbation is considered to be a bounded transfer function (with respect to its \( \infty \)-norm).
- This type of uncertainty is referred to as unstructured since no detailed model of perturbation is employed.

Types of unstructured uncertainty

- Additive (unknown dynamics in parallel with the plant)
  \[ G(s) = G_o(s) + \Delta_a(s). \]

- Input multiplicative (unknown dynamics in series with the plant)
  \[ G(s) = G_o(s)[I + \Delta_i(s)]. \]

- Output multiplicative (unknown dynamics in series with the plant)
  \[ G(s) = [I + \Delta_o(s)]G_o(s). \]

- Input feedback (uncertainty in gain/phase/pole locations of plant)
  \[ G(s) = G_o(s)[I - \Delta_{fi}(s)]^{-1}. \]

- Output feedback (uncertainty in gain/phase/pole locations of plant)
  \[ G(s) = [I - \Delta_{fo}(s)]^{-1}G_o(s). \]
- Analysis may be performed when model perturbations are bounded

\[ \tilde{\sigma}\{\Delta'\} \leq \Delta_{\max}(j\omega) \]

where \( \tilde{\sigma} \) is the maximum singular value and \( \Delta' \) is any of perturbation types considered.

- The bound \( \Delta_{\max} \) is generally frequency-dependent, so uncertainty may vary over frequency.

- All the unstructured uncertainty models may be analyzed in a similar manner by placing them in a common framework.

- The perturbation is normalized so \( \|\Delta(j\omega)\|_\infty \leq 1 \) by defining

\[ \Delta(j\omega) = \frac{1}{\Delta_{\max}(j\omega)} \Delta'(j\omega) \]

and

\[
\begin{bmatrix}
Y_d(s) \\
Y(s) \\
M(s)
\end{bmatrix}
= \begin{bmatrix}
P_{yd}(s) & P_{yd}(s) & P_{yu}(s) \\
P_{yd}(s) & P_{yd}(s) & P_{yu}(s) \\
P_{yd}(s) & P_{yd}(s) & P_{yu}(s)
\end{bmatrix}
\begin{bmatrix}
W_d(s) \\
W(s) \\
U(s)
\end{bmatrix}
= P(s)
\begin{bmatrix}
W_d(s) \\
W(s) \\
U(s)
\end{bmatrix}.
\]

- The “plant” \( P(s) \) is generally a combination of \( G_o(s) \) and uncertainties added together.

**EXAMPLE:** We consider a plant model to have input multiplicative uncertainty. The nominal model is accurate to within about 0.5\% at frequencies below 10 \( \text{rad s}^{-1} \) but is inaccurate to within about 50\% at frequencies above 1000 \( \text{rad s}^{-1} \). The accuracy should transition between these two extremes at intermediate frequencies.
We can model the uncertainty as a first-order transfer function with a zero at 10 rad s\(^{-1}\) and a pole at 1000 rad s\(^{-1}\).

\[
\Delta'_{\text{max}}(j\omega) = 0.5 \frac{(j\omega + 10)}{(j\omega + 1000)}.
\]

The uncertainty is coupled into standard form as:

From the diagram we see

\[
P(s) = \begin{bmatrix}
0 & \frac{0.5(s + 10)}{s + 1000} & -\frac{0.5(s + 10)}{s + 1000} \\
\frac{G_o(s)}{s + 1000} & \frac{G_o(s)}{s + 1000} & -G_o(s) \\
\frac{G_o(s)}{s + 1000} & \frac{G_o(s)}{s + 1000} & -G_o(s)
\end{bmatrix}.
\]

**Modeling structured uncertainty**

Unstructured uncertainty is modeled by connecting unknown but bounded perturbations to the plant.
• Sometimes, more information is available ⇒ New constraints on uncertainty add “structure” to the set of admissible perturbations.
  • Plant subject to multiple perturbations.
  • Plant has multiple uncertain parameters.
  • Multiple unstructured but independent uncertainties.

Structured uncertainty model similar to before

\[
\Delta(s) = \begin{bmatrix}
\Delta_1(s) & 0 & \cdots & 0 \\
0 & \Delta_2(s) & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \Delta_n(s)
\end{bmatrix}
\]

is block diagonal.

• An individual \( \Delta_k(s) \) may represent an uncertain parameter (scalar) or an unstructured uncertainty.

• All blocks have \( \|\Delta_k\|_{\infty} \leq 1 \).

**EXAMPLE:** Consider the transfer function

\[
G(s) = \frac{1}{(s + p_1)(s + p_2)}.
\]

• The two poles are uncertain, as given by

\[
p_1 \in [0.9, \, 1.1]; \quad p_2 \in [3, \, 5].
\]

• We can model the poles as nominal values plus perturbation

\[
p_1 = 1 + \delta_1; \quad p_2 = 4 + \delta_2
\]

where \(-0.1 \leq \delta_1 \leq 0.1\) and \(-1 \leq \delta_2 \leq 1\). We can place the perturbations in a feedback loop around the nominal plant as
The perturbations can be normalized and the system put into standard form such that

\[ \| \Delta_1 \|_\infty \leq 1; \quad \| \Delta_2 \|_\infty \leq 1; \quad \| \Delta \|_\infty \leq 1. \]
7.5: Four classic problems

- Will the techniques we have already learned work well if plant dynamics are uncertain? → Analysis.
- How do we design controllers to account for the uncertainty explicitly? → Synthesis.

Nominal stability

- Controller stabilizes nominal system $G_o(s)$.
- Will generalize notion of stability to internal stability.
- Consider the following diagram.

![Diagram](image)

Consider

$$\begin{bmatrix} E_1(s) \\ E_2(s) \end{bmatrix} = \begin{bmatrix} (I + KG_o)^{-1} & -(I + KG_o)^{-1}K \\ (I + G_o K)^{-1}G_o & (I + G_o K)^{-1} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix},$$

which are the transfer functions from the inputs to the “errors”. (equivalent to transfer functions from inputs to outputs).

- System is internally stable if all four transfer functions are stable.
- Nominal stability a relatively easy problem. For example, you might use pole-placement techniques from ECE4520/5520.

Robust stability

- More interesting because we acknowledge uncertainty in our model.
Now we must define what we think is wrong with the model ➞ Use structured or unstructured uncertainty models.

Define $\Gamma$ to be the set of all admissible $G(s)$ given the nominal $G_o(s)$ and uncertainty model.

**ANALYSIS:** Does $K(s)$ internally stabilize all $G(s) \in \Gamma$?

**SYNTHESIS:** Design the controller $K(s)$ to maximize $\gamma$ where

$$\|\Delta(j\omega)\|_\infty \leq \gamma \quad \text{Make the system more robust to the types of uncertainty expected.}$$

- No other performance objectives specified.

**Nominal performance**

- Decide on an objective. *e.g.*, disturbance rejection, reference tracking.
- Develop assumptions about inputs. *e.g.*, $r(t)$ is a step or ramp; $w(t)$ is stochastic WGN with known covariance.
- Characterize performance metric. *e.g.*, minimize the LQG cost function.
- For example, if $d$ is some kind of disturbance input,

$$K(s) = \arg\min_{\mathcal{H}} \left\{ \max_d \left[ \text{energy}(y) \right] \right\}$$

with constraints on $\text{energy}(d)$. (minimize the worst case over $d$).

**Robust performance**

- Design $K(s)$ so that internal stability and performance of a specified type holds for all $G(s) \in \Gamma$. 
Disturbance rejection must be good for all possible $G(s)$.

$$K(s) = \arg \min_{\mathcal{H}} \left\{ \max_{d, \Delta} \left[ \text{energy}(y) \right] \right\}$$

with constraints on energy($d$). (minimize the worst-case over $d, \Delta$).

VERY VERY hard for some types of uncertainty.
7.6: Robust stability [analysis for unstructured uncertainty]

- We want to determine whether our controlled system is stable for all admissible plant perturbations $\Delta \in \tilde{\Delta}$.

- Start with the 3-input 3-output standard form

\[
\begin{bmatrix}
Y_d(s) \\
Y(s) \\
M(s)
\end{bmatrix} = \begin{bmatrix}
P_{yd}w_d(s) & P_{yd}w(s) & P_{yd}u(s) \\
P_{yw_d}(s) & P_{yw}(s) & P_{yu}(s) \\
P_{mw_d}(s) & P_{mw}(s) & P_{mu}(s)
\end{bmatrix} \begin{bmatrix}
W_d(s) \\
W(s) \\
U(s)
\end{bmatrix} = P(s) \begin{bmatrix}
W_d(s) \\
W(s) \\
U(s)
\end{bmatrix}.
\]

- When a controller is added to the system,

\[
U(s) = K(s)M(s)
\]

\[
= K(s)P_{mw_d}(s)W_d(s) + K(s)P_{mw}(s)W(s) + K(s)P_{mu}(s)U(s)
\]

\[
= \{I - K(s)P_{mu}(s)\}^{-1}K(s)P_{mw_d}(s)W_d(s) +
\]

\[
\{I - K(s)P_{mu}(s)\}^{-1}K(s)P_{mw}(s)W(s).
\]

- We get the following closed-loop system (dropping “s” for clarity):

\[
\begin{bmatrix}
Y_d \\
Y
\end{bmatrix} = \begin{bmatrix}
N_{yd}w_d & N_{yd}w \\
N_{yw_d} & N_{yw}
\end{bmatrix} \begin{bmatrix}
W_d \\
W
\end{bmatrix},
\]

where

\[
N_{yd}w_d = P_{yd}w_d + P_{yd}u\{I - KP_{mu}\}^{-1}KP_{mw_d}
\]

\[
N_{yd}w = P_{yd}w + P_{yd}u\{I - KP_{mu}\}^{-1}KP_{mw}
\]

\[
N_{yw_d} = P_{yw_d} + P_{yu}\{I - KP_{mu}\}^{-1}KP_{mw_d}
\]

\[
N_{yw} = P_{yw} + P_{yu}\{I - KP_{mu}\}^{-1}KP_{mw}.
\]
The modified standard form, incorporating the controller dynamics, is:

- The nominal system is assumed to be stable.
- The perturbation is also assumed to be stable.
- The combined system is then stable iff the feedback loop around the perturbation is internally stable.
- We evaluate the robust internal stability of the system by considering the feedback loop

\[
\begin{align*}
U_1(s) & \quad + \quad E_1(s) \quad \rightarrow \quad Y_1(s) \\
Y_2(s) & \quad \rightarrow \quad E_2(s) \quad + \quad U_2(s)
\end{align*}
\]

- The transfer functions defined by the system are:

\[
\begin{pmatrix}
E_1 \\
E_2
\end{pmatrix}
= \begin{pmatrix}
(I - \Delta N_{y_d w_d})^{-1} & (I - \Delta N_{y_d w_d})^{-1} \\
(I - \Delta N_{y_d w_d})^{-1} N_{y_d w_d} & (I - \Delta N_{y_d w_d})^{-1}
\end{pmatrix}
\begin{pmatrix}
U_1 \\
U_2
\end{pmatrix}.
\]

- Consider the first transfer function

\[
E_1(s) = (I - \Delta(s) N_{y_d w_d}(s))^{-1} U_1(s) = \Delta(s) N_{y_d w_d}(s) E_1(s) + U_1(s).
\]

- In the time domain, \((\delta(t) \text{ is } L^{-1}(\Delta(s)))\), not the impulse function!

\[
e_1(t) = \delta(t) * n_{y_d w_d}(t) * e_1(t) + u_1(t).
\]

- This part of the system is stable if a bounded input produces a bounded output. Take the 2-norm of the signals to see:
\[
\|e_1(t)\|_2 = \|\delta(t) * n_{yd \, wd}(t) * e_1(t) + u_1(t)\|_2 \\
\leq \|\delta(t) * n_{yd \, wd}(t) * e_1(t)\|_2 + \|u_1(t)\|_2.
\]

- We can use the $\infty$-norm to bound the terms on the right-hand-side

\[
\|e_1(t)\|_2 \leq \|\Delta N_{yd \, wd}\|_\infty \|e_1(t)\|_2 + \|u_1(t)\|_2 \\
\leq \|\Delta\|_\infty \|N_{yd \, wd}\|_\infty \|e_1(t)\|_2 + \|u_1(t)\|_2 \\
= (1 - \|\Delta\|_\infty \|N_{yd \, wd}\|_\infty)^{-1} \|u_1(t)\|_2.
\]

- This is stable if the inverse is finite. Since $\|\Delta\|_\infty \leq 1$ then the condition for stability is

\[
\|N_{yd \, wd}\|_\infty < 1.
\]

- The book examines the transfer function $E_1/U_2$. All three other transfer functions have exactly the same condition for robust stability.

**CONCLUSION:** System is robustly stable iff

\[
\|N_{yd \, wd}\|_\infty = \sup_{\omega} \{\tilde{\sigma}[N_{yd \, wd}(j\omega)]\} < 1.
\]

**EXAMPLE:** We are given the nominal plant

\[
G_o(s) = \frac{10}{s - 1},
\]

with unstructured additive uncertainty

\[
\|\Delta(s)\|_\infty \leq 1,
\]

and controller $K(s) = -\frac{1}{2}$. 

We can compute

\[ N_{yd wd}(s) = P_{yd wd} + P_{yd u}(I - KP_{mu})^{-1}KP_{mw d} \]

\[ = 0 + (1) \left(1 - \left(-\frac{1}{2}\right) \left(\frac{10}{s - 1}\right)\right)^{-1} \left(-\frac{1}{2}\right) \]

\[ = \left(\frac{s + 4}{s - 1}\right)^{-1} \left(-\frac{1}{2}\right) = \frac{-1/2(s - 1)}{s + 4}. \]

The nominal closed-loop system has a pole at \( s = -4 \) (stable). Is the system robustly stable?

Compute

\[ \tilde{\sigma}\{N_{yd wd}(j\omega)\} = |N_{yd wd}(j\omega)| = \frac{1}{2} \frac{\sqrt{\omega^2 + 1}}{\sqrt{\omega^2 + 16}}. \]

The \( \infty \)-norm is (the max is at \( \omega \to \infty \))

\[ \| N_{yd wd}(j\omega) \|_\infty = \sup_{\omega} \frac{1}{2} \frac{\sqrt{\omega^2 + 1}}{\sqrt{\omega^2 + 16}} = \frac{1}{2}. \]

Therefore the system is robustly stable.
7.7: Robust stability [analysis for structured uncertainty]

- When the uncertainty has structure, the robust stability test  
  \[ \| N_{y_d w_d} \|_\infty < 1 \]  
  is overly conservative.

- It assumes that \( \Delta \) may have any structure subject to  
  \[ \| \Delta \|_\infty \leq 1 \]. A prediction of instability may be based on a specific \( \Delta \) that is not allowed given its structure.

- When there is structure, we revert back to the internal stability test and notice that the closed-loop system becomes unstable for \( \Delta \) such that
  \[ \det \{ I - N_{y_d w_d} (s) \Delta (s) \} = 0. \]

- To determine robust stability, we solve for the “smallest” delta that makes this true
  \[ \inf_{\Delta(j\omega) \in \tilde{\Delta}} \left\{ \min_{\Delta(j\omega) \in \tilde{\Delta}} \{ \bar{\sigma} [\Delta(j\omega)] : \det \{ I - N_{y_d w_d} (j\omega) \Delta (j\omega) = 0 \} \} \right\}. \]

- If this value is greater than 1 (the maximum size of \( \Delta \)) then the system is robustly stable.

- Easier to solve if we define
  \[ \mu_{\tilde{\Delta}}(N) = \frac{1}{\min_{\Delta \in \tilde{\Delta}} \{ \bar{\sigma} [\Delta] : \det \{ I - N_{y_d w_d} \Delta \} = 0 \}}. \]

- \( \mu_{\tilde{\Delta}} \) is called the “Structured Singular Value”.

- Then, the system is robustly stable iff
  \[ \sup_{\omega} \{ \mu_{\tilde{\Delta}} [N_{y_d w_d} (j\omega)] \} < 1. \]

**Computing \( \mu_{\tilde{\Delta}}(N_{y_d w_d}) \).**

- The bad news: There is no closed-form solution or numeric algorithm to compute \( \mu_{\tilde{\Delta}}(N_{y_d w_d}) \) in the general case.
The better news: We can often find quite good bounds on it. In particular, we know that
\[
\mu_{\Delta}(N_{ydwd}) \leq \sigma(N_{ydwd}).
\]

This should be obvious since the structured singular value is a special case of the regular singular value.

To make this bound better, consider:

\[
\begin{bmatrix}
    d_1(s)I_{l_1} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & d_n(s)I_{l_n}
\end{bmatrix} \rightarrow \begin{bmatrix}
    \Delta_1(s) & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & \Delta_n(s)
\end{bmatrix} \rightarrow \begin{bmatrix}
    \frac{1}{d_1(s)}I_{n_1} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & \frac{1}{d_n(s)}I_{n_n}
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \frac{1}{d_1(s)}I_{l_1} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & \frac{1}{d_n(s)}I_{l_n}
\end{bmatrix} \rightarrow \frac{1}{d_1(s)}I_{n_1} \rightarrow \begin{bmatrix}
    d_1(s)I_{n_1} & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & d_n(s)I_{n_n}
\end{bmatrix}
\]

or

\[
\begin{bmatrix}
    \Delta_1(s) & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & \Delta_n(s)
\end{bmatrix} \rightarrow \mathcal{D}_R(s)N_{ydwd}(s)\mathcal{D}_L^{-1}(s)
\]

We see that
$$\mu_{\bar{\Delta}}(N(s)) = \mu_{\bar{\Delta}}[\mathcal{D}_R(s)N_{ydwd}(s)\mathcal{D}_L^{-1}(s)] \leq \bar{\sigma}[\mathcal{D}_R(s)N_{ydwd}(s)\mathcal{D}_L^{-1}(s)],$$

for any $\mathcal{D}_R$ and $\mathcal{D}_L$.

- In particular,

$$\mu_{\bar{\Delta}}(N_{ydwd}(s)) \leq \min_{\{d_1, \ldots, d_n\}} \bar{\sigma}[\mathcal{D}_R(s)N_{ydwd}(s)\mathcal{D}_L^{-1}(s)],$$

- This minimization is a convex-optimization problem for which a unique minima exists, and very efficient algorithms exist to solve it. The final bound is usually very close to the true structured-singular value.

**EXAMPLE:** Compute the $\mu$ value for

$$N_{ydwd}(s) = \begin{bmatrix} \frac{1}{s+1} & \frac{1}{s+1} \\ \frac{s+1}{s+1} & \frac{s+1}{s+1} \end{bmatrix}.$$

- Here, we use the optimization toolbox, but you could code your own optimization routine (gradient descent, for example).

```matlab
function mustuff
    global Nydwd
dscale=[1;1]; % initial guess for dscale
w=logspace(-2,2,100);
for i=1:100,
    Nydwd=[1/(1j*w(i)+1) 1/(1j*w(i)+1); 1j*w(i)/(1j*w(i)+1) 1j*w(i)/(1j*w(i)+1)];
    s(i)=max(svd(Nydwd));
    dscale=fminunc(@estimmu,dscale);
    mui(i)=estimmu(dscale);
end
max(s),max(mui),semilogx(w,s,w,mui);
return

function muout = estimmu(d)
    global Nydwd
    muout = max(svd(diag(d)*Nydwd*diag(1./d)));
return
```
Maximum singular value and $\mu$ value

![Graph showing gain vs. frequency for maximum singular value and $\mu$ value.](image-url)
7.8: Performance robustness [analysis]

- Robust performance is defined in different ways than we are accustomed to.
- Bounds are placed on certain performance variables, such as steady-state error, disturbance in the output, control effort.
- These bounds are scaled in standard form, such that the transfer function between the reference- and disturbance inputs and the performance outputs must be bounded by infinity-norm 1:

\[ \| T_{yw} \|_{\infty} < 1, \]

where \( T_{yw} \) is the closed-loop transfer function between the inputs and performance variables.

- The closed loop transfer function is

\[ T_{yw} = N_{yw_d}(s)[I - \Delta(s)N_{yd_d}(s)]^{-1}\Delta(s)N_{yd}(s) + N_{yw}(s). \]

- We can visualize this as a structured-singular-value stability problem, where we add an additional “uncertainty” block.

**EXAMPLE:** Consider a plant with uncertain pole location:

\[ G(s) = \frac{1}{s + 1 + \delta}; \quad \delta \in \{-0.2, 0.2\}. \]

- A proportional-gain controller \( K \) is to be used for this system.
■ The reference input is bandlimited to less than $10 \text{ rad s}^{-1}$. Tracking error should be less than 0.1.

■ We can then bound the closed-loop transfer function from reference input to error output.

$$|T_{yw}(j\omega)| \leq \begin{cases} 0.1, & \omega \leq 10; \\ \infty, & \omega > 10. \end{cases}$$

■ The $\infty$ indicates that error in frequencies above $10 \text{ rad s}^{-1}$ is unimportant.

■ We normalize the performance goal using a weighting function

$$W(j\omega) = \begin{cases} 10, & \omega \leq 10; \\ 0, & \omega > 10. \end{cases}$$

■ A low-order rational approximation of this weighting function is

$$W(j\omega) = \frac{150}{j\omega + 10}.$$ 

Robust performance can be analyzed by using the SSV by appending the weighting function to the plant and adding the performance block.
The transfer function of the system is

\[
N(s) = \begin{bmatrix}
-0.2 & K \\
\frac{s + 1 + K}{30} & \frac{s + 1 + K}{150(s + 1)} \\
\frac{(s + 1 + K)(s + 10)}{(s + 1 + K)(s + 10)} & \frac{(s + 1 + K)(s + 10)}{(s + 1 + K)(s + 10)}
\end{bmatrix}
\]

Robust performance is tested by generating the SSV of \( N(j\omega) \). The system is stable and meets the performance specifications for all admissible perturbations if SSV is less than 1 for all frequencies.

**Design of robust controllers**

- Design of controllers to achieve robust stability and performance is challenging.
- Instead of minimizing quadratic cost functions (\( H_2 \) control), \( \infty \)-norm cost functions are minimized (\( H_\infty \) control).
- Numeric optimization methods are required \( \Rightarrow \) No closed-form or algorithmic solution is available. A sub-optimal controller is generally obtained, but is accepted.
- The subject of chapters 9–11 in the text.
We will encounter familiar concepts such as:

- Cost functions,
- Optimization via calculus of variations,
- Riccati equations,
- Hamiltonian systems,
- Regulators, Estimators and so forth.
Appendix: Doyle code

% DOYLE.M - Demonstrate LQG lack of robustness w/ Doyle's example.

A=[1 1; 0 1]; Bu=[0; 1]; Cy=[1 0];
f=logspace(-8,3,500); s=j*f;

Q=ones(2); Sw=Q;
R=1; Sv=R;

[K,P]=lqr(A,Bu,Q,R);
[L,Sigmae]=lqr(A',Cy',Sw,Sv); L=L';

Acl=A-Bu*K-L*Cy; Bcl=L; Ccl=K; Dcl=0;

Glqr = freqresp(A,Bu,K,0,1,s); % LQR system freqresp
Gcl = freqresp(Acl,Bcl,Ccl,Dcl,1,s); % COMPENSATOR freqresp
Gp = freqresp(A,Bu,Cy,0,1,s); % PLANT freqresp

circ=exp(pi*s/max(abs(s)));

clf; subplot(131);
plot(-1,0,'r+','markersize',18); hold on;
plot(Gp.*Gcl); plot(conj(Gp.*Gcl)); % Nyquist LQG
plot(circ,'-.'); plot(-circ,'-.');
plot([-1+circ,:]); plot([-1-circ,:]);
plot([-1 0],[-2 2],'y--'); plot([-2 2],[0 0],'y--');
axis([-1.25 .25 -.75 .75]); axis('square')
text(-.5,.5,['R = ',num2str(R)]);
plot(-1,0,'r+','markersize',18);
title('Doyle Example: LQG Nyquist'); xlabel('Real'); ylabel('Imag');

subplot(132);
plot(-1,0,'r+','markersize',18); hold on;
plot(Gp.*Gcl); plot(conj(Gp.*Gcl)); % Nyquist LQG
plot(circ,'-.'); plot(-circ,'-.');
plot([-1+circ,:]); plot([-1-circ,:]);
plot([-1 0],[-2 2],'y--'); plot([-2 2],[0 0],'y--');
axis([-1.1 -.9 -.1 .1]); axis('square');
plot(-1,0,'r+','markersize',18);
title('Doyle Example: LQG Nyquist (zoom)'); xlabel('Real'); ylabel('Imag');
```matlab
subplot(133);
plot(-1,0,'r+','markersize',18); hold on;
plot(GLqr); plot(conj(GLqr));
plot(circ,'-'); plot(-circ,'-');
plot(-1+circ,':'); plot(-1-circ,':');
plot([0 0],[-2 2],'y--'); plot([-3 2],[0 0],'y--');
axis([-2.5 .5 -1.5 1.5]); axis('square')
text(-1.5,.5,['R = ',num2str(R)]);
plot(-1,0,'r+','markersize',18);
title('Doyle Example: LQR Nyquist'); xlabel('Real'); ylabel('Imag');
```