Topic #19

16.31 Feedback Control Systems

- Stengel Chapter 6
- Question: how well do the large gain and phase margins discussed for LQR map over to DOFB using LQR and LQE (called LQG)?

Cite as: Jonathan How, course materials for 16.31 Feedback Control Systems, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

Linear Quadratic Gaussian (LQG)

- When we use the combination of an optimal estimator and an optimal regulator to design the controller, the compensator is called

  Linear Quadratic Gaussian (LQG)

  – Special case of the controllers that can be designed using the separation principle.

- Great news about an LQG design is that stability of the closed-loop system is **guaranteed**.

  – The designer is freed from having to perform any detailed mechanics - the entire process is fast and automated.

  – Designer can focus on the "performance" related issues, being confident that the LQG design will produce a controller that stabilizes the system.

  ◇ Selecting values of $R_{\rm zz}$, $R_{\rm uu}$ and relative sizes of $R_{\rm ww}$ & $R_{\rm vv}$

- This sounds great – so what is the catch??

- Remaining issue is that sometimes the controllers designed using these state space tools are very sensitive to errors in the knowledge of the model.

  – *i.e.,* the compensator might work **very well** if the plant gain $\alpha = 1$, but be unstable if $\alpha = 0.9$ or $\alpha = 1.1$.

  – LQG is also prone to plant–pole/compensator–zero cancelation, which tends to be sensitive to modeling errors.

- J. Doyle, "Guaranteed Margins for LQG Regulators", IEEE *Transactions on Automatic Control*, Vol. 23, No. 4, pp. 756-757, 1978.

- The good news is that the state-space techniques will give you a controller very easily.
  - **You should use the time saved to verify that the one you designed is a "good" controller.**

- There are, of course, different definitions of what makes a controller **good**, but one important criterion is whether **there is a reasonable chance that it would work on the real system as well as it does in Matlab.** ⇒ **Robustness**.
  - The controller must be able to tolerate some modeling error, because our models in Matlab are typically inaccurate.
    - ◇ Linearized model
    - ◇ Some parameters poorly known
    - ◇ Ignores some higher frequency dynamics

- Need to develop tools that will give us some insight on how well a controller can tolerate modeling errors.

Cite as: Jonathan How, course materials for 16.31 Feedback Control Systems, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

Example

- Consider the “cart on a stick” system, with the dynamics as given in the following pages. Define

$$q = \begin{bmatrix} \theta \\ x \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}$$

  Then with $y = x$

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B_u u \\ y &= C_y \mathbf{x} \end{aligned}$$

- For the parameters given in the notes, the system has an unstable pole at $+5.6$ and one at $s = 0$. There are plant zeros at ± 5.
- Very simple LQG design - main result is fairly independent of the choice of the weighting matrices.
- The resulting compensator is unstable (+23!!)
  - This is somewhat expected. (why?)

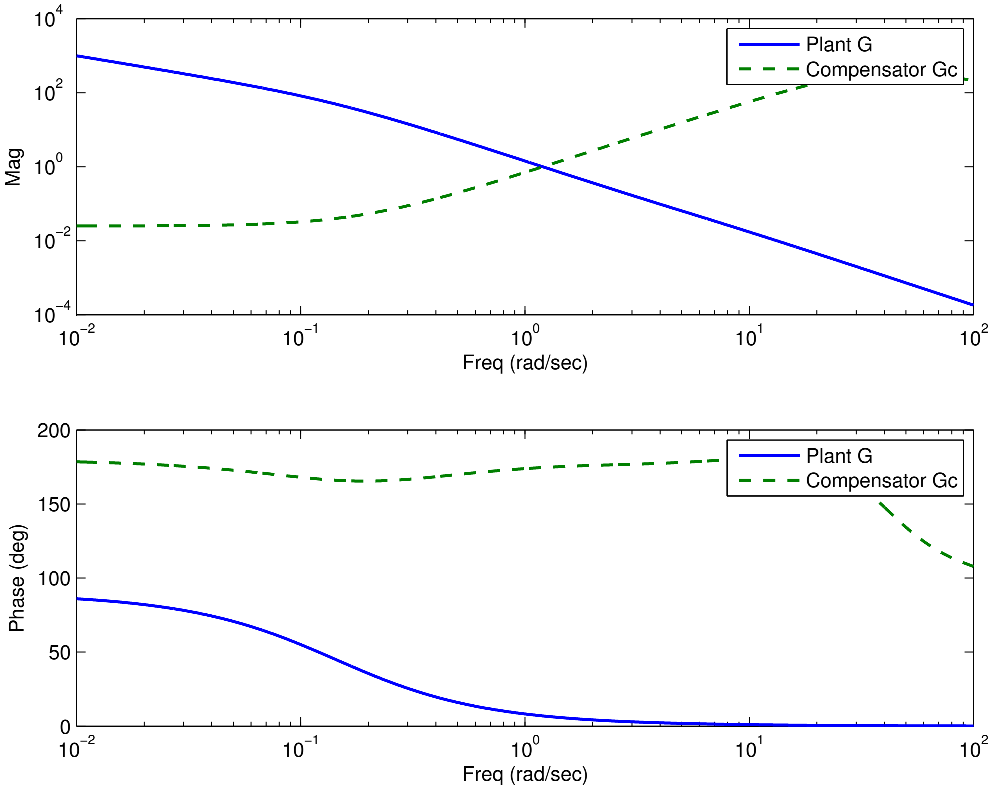


Figure 1: Plant and Controller

Example: cart with an inverted pendulum.

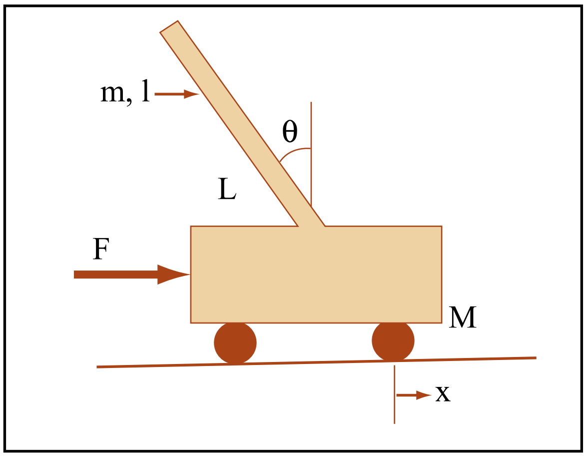


Figure by MIT OpenCourseWare.

- Nonlinear equations of motion can be developed for large angle motion (see 30-32)
- Force actuator, θ sensor

Linearize for small θ

(I+mL2) θ

$$\begin{bmatrix} (I+mL^2)s^2 - mgL & -mLs^2 \\ -mLs^2 & (M+m)s^2 + Gs \end{bmatrix} \begin{bmatrix} \Theta(s) \\ x(s) \end{bmatrix} = \begin{bmatrix} 0 \\ F(s) \end{bmatrix}$$

$$\frac{\Theta}{F} = \frac{mLs^2}{\left[(I+mL^2)s^2 - mgL\right]\left[(M+m)s^2 + Gs\right] - (mLs^2)^2}$$

Cannot say too much more

Let M= 0.5, m=0.2, G=0.1, I=0.006, L=0.3

➔ gives $\frac{\Theta}{F} = \frac{4.54s^2}{s^4 + 0.1818s^3 - 31.18s^2 - 4.45s}$

therefore has an unstable pole (as expected)
s=± 5.6,-0.14,0

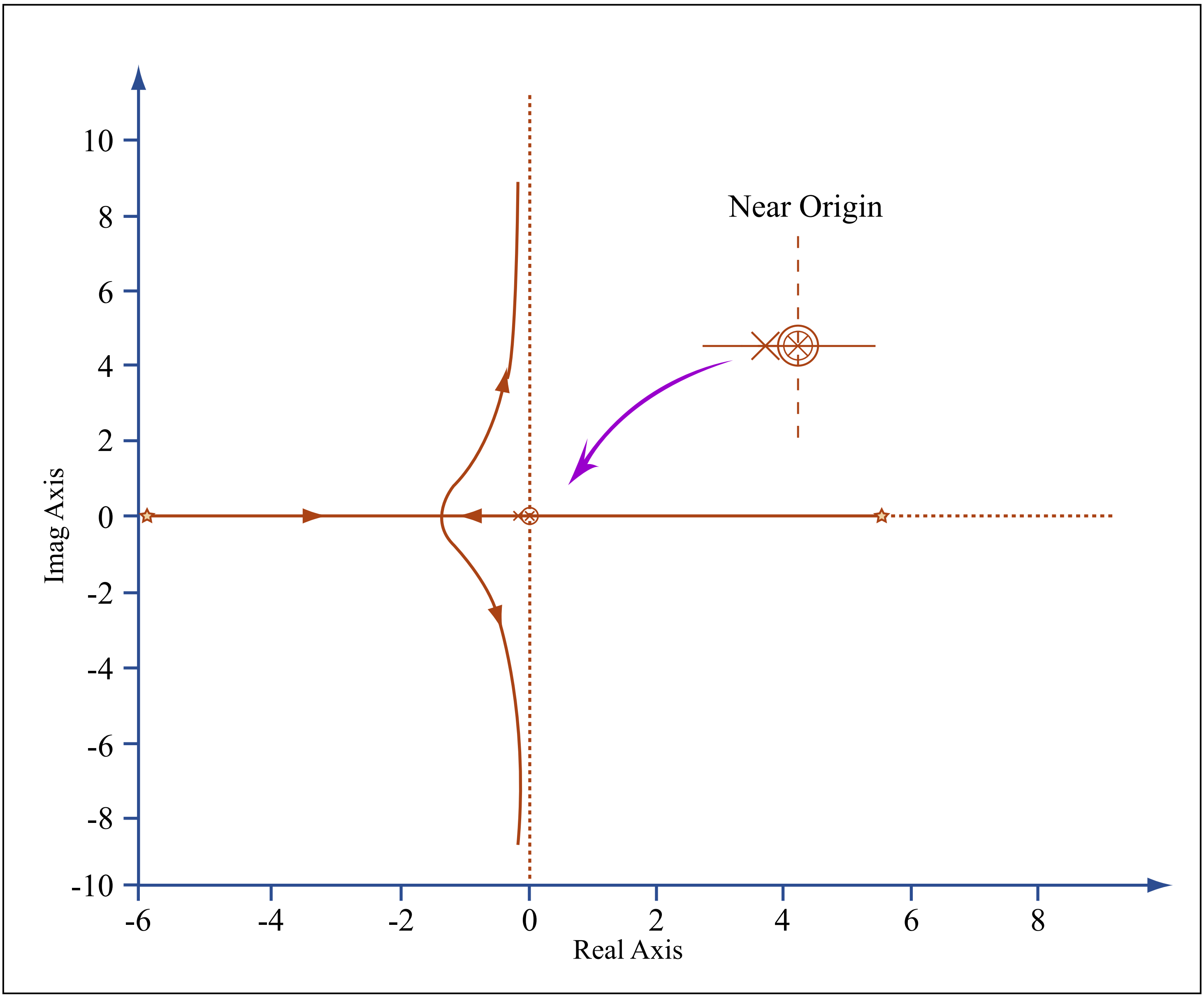


Figure by MIT OpenCourseWare.

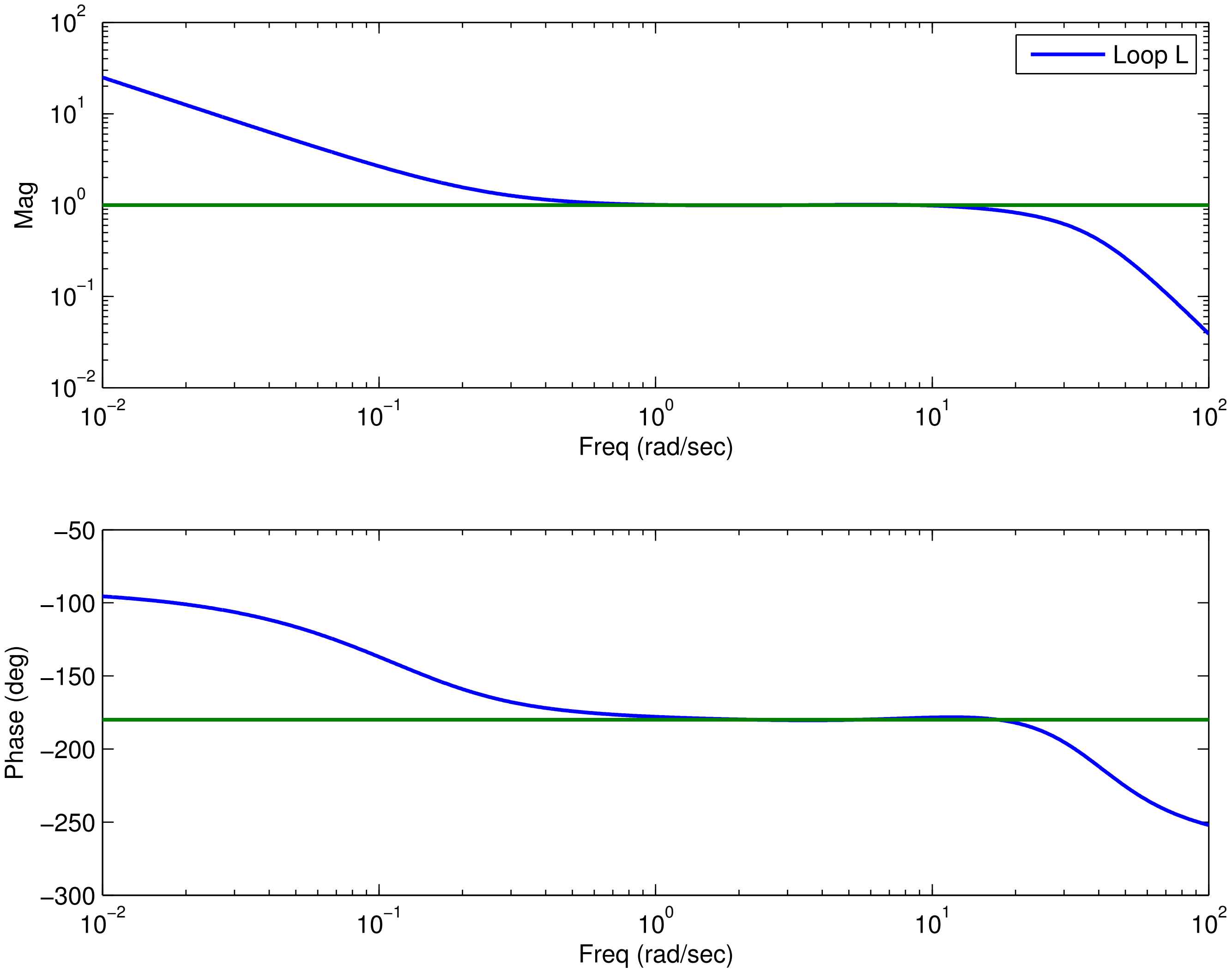


Figure 2: Loop and Margins

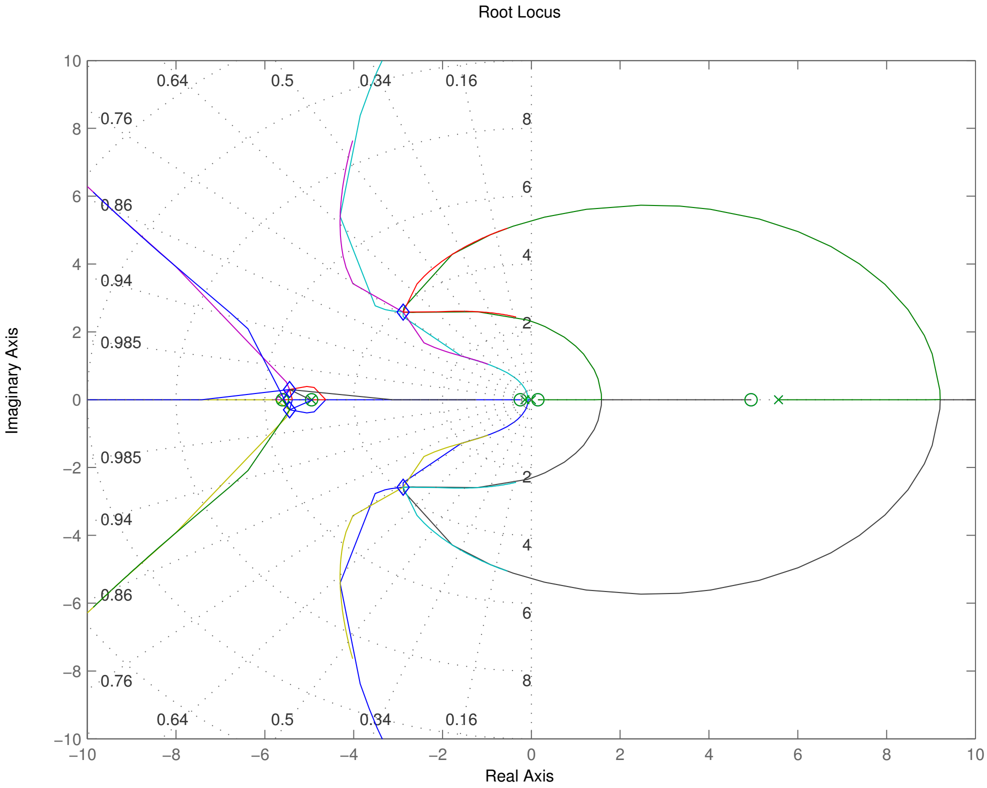
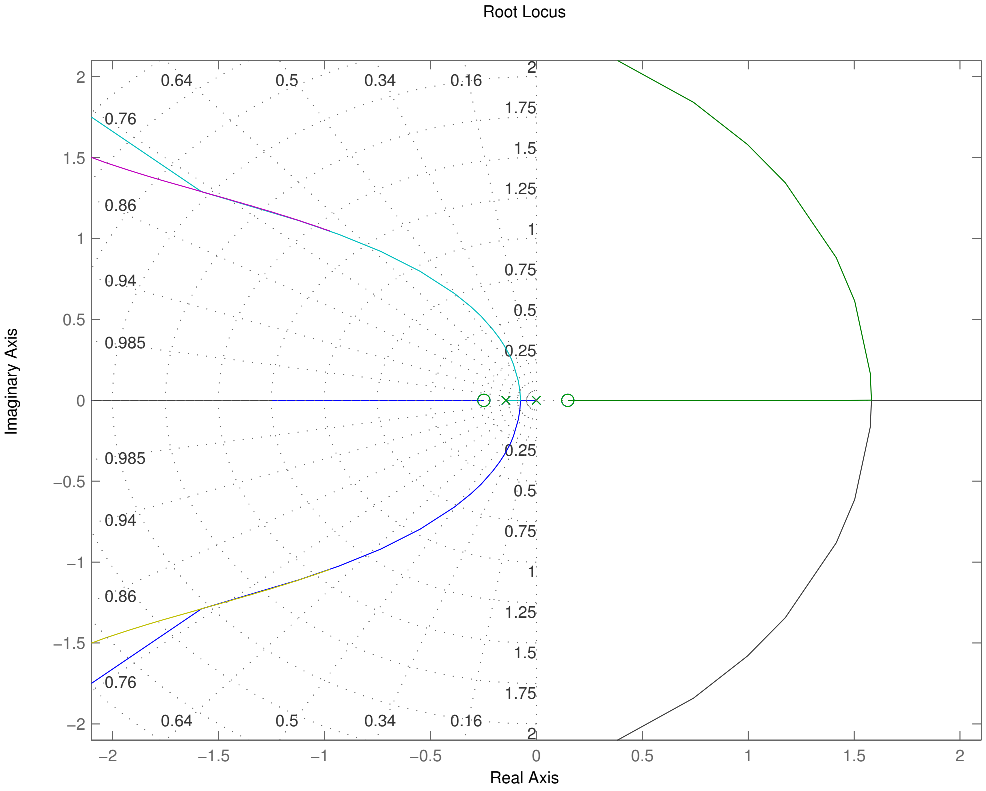


Figure 3: Root Locus with frozen compensator dynamics. Shows sensitivity to overall gain – symbols are a gain of [0.995:.0001:1.005].



- Looking at both the Loop TF plots and the root locus, it is clear this system is stable with a gain of 1, but
  - Unstable for a gain of $1 \pm \epsilon$ and/or a slight change in the system phase (possibly due to some unmodeled delays)
  - Very limited chance that this would work on the real system.

- Of course, this is an extreme example and not all systems are like this, but you must analyze to determine what **robustness margins** your controller really has.

- **Question:** what analysis tools should we use?

Cite as: Jonathan How, course materials for 16.31 Feedback Control Systems, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

Analysis Tools to Use?

- Eigenvalues give a definite answer on the stability (or not) of the closed-loop system.
  - Problem is that it is very hard to predict where the closed-loop poles will go as a function of errors in the plant model.

- Consider the case were the model of the system is

$$\dot{\mathbf{x}} = A_0\mathbf{x} + B\mathbf{u}$$

  - Controller also based on A_0, so **nominal** closed-loop dynamics:

$$\begin{bmatrix} A_0 & -BK \\ LC & A_0 - BK - LC \end{bmatrix} \Rightarrow \begin{bmatrix} A_0 - BK & BK \\ 0 & A_0 - LC \end{bmatrix}$$

- But what if the **actual** system has dynamics

$$\dot{\mathbf{x}} = (A_0 + \Delta A)\mathbf{x} + B\mathbf{u}$$

  - Then **perturbed** closed-loop system dynamics are:

$$\begin{bmatrix} A_0 + \Delta A & -BK \\ LC & A_0 - BK - LC \end{bmatrix} \Rightarrow \begin{bmatrix} A_0 + \Delta A - BK & BK \\ \Delta A & A_0 - LC \end{bmatrix}$$

- Transformed $\bar{A}_{cl}$ not upper-block triangular, so perturbed closed-loop eigenvalues are **NOT** the union of regulator & estimator poles.
  - Can find the closed-loop poles for a specific ΔA, but
  - Hard to predict change in location of closed-loop poles for a range of possible modeling errors.

Frequency Domain Tests

- Frequency domain stability tests provide further insights on the **stability margins**.

- Recall that the **Nyquist Stability Theorem** provides a binary measure of stability, or not.

- But already discussed that we can use "closeness" of $L(s)$ to the critical point as a measure of "closeness" to changing the number of encirclements.
  - Closeness translates to high sensitivity which corresponds to $L_N(\mathbf{j}\omega)$ being **very** close to the critical point.
  - Ideally you would want the sensitivity to be low. Same as saying that you want $L(\mathbf{j}\omega)$ to be far from the critical point.

- Premise is that the system is stable for the nominal system
  $\Rightarrow$ has the right number of encirclements.
  - Goal of the robustness test is to see if the possible perturbations to our system model (due to modeling errors) can **change the number of encirclements**
  - In this case, say that the perturbations can **destabilize** the system.

Cite as: Jonathan How, course materials for 16.31 Feedback Control Systems, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].

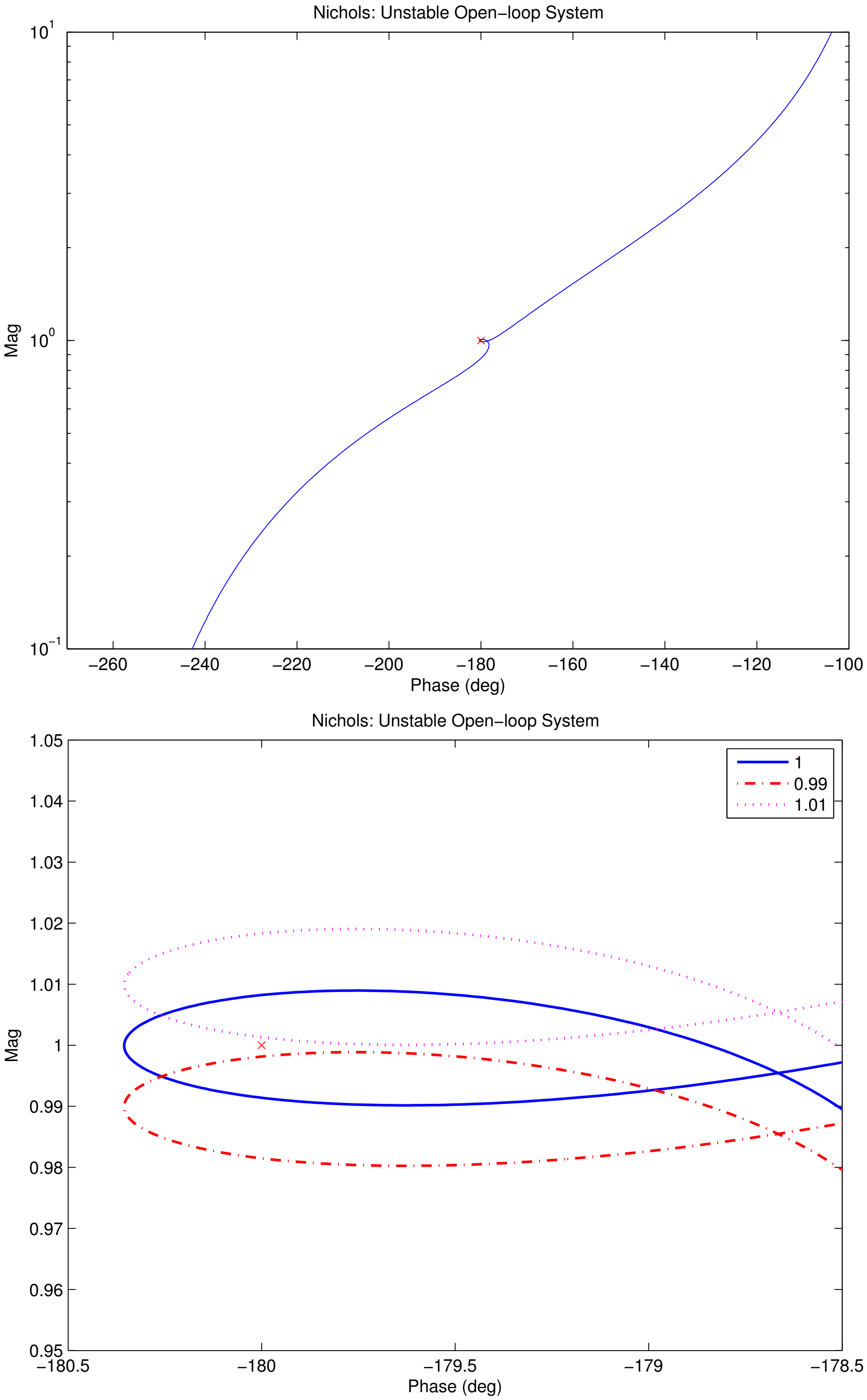


Figure 4: Nichols Plot ($|L((\mathbf{j}\omega))|$ vs. $\arg L((\mathbf{j}\omega))$) for the cart example which clearly shows sensitivity to overall gain and/or phase lag.

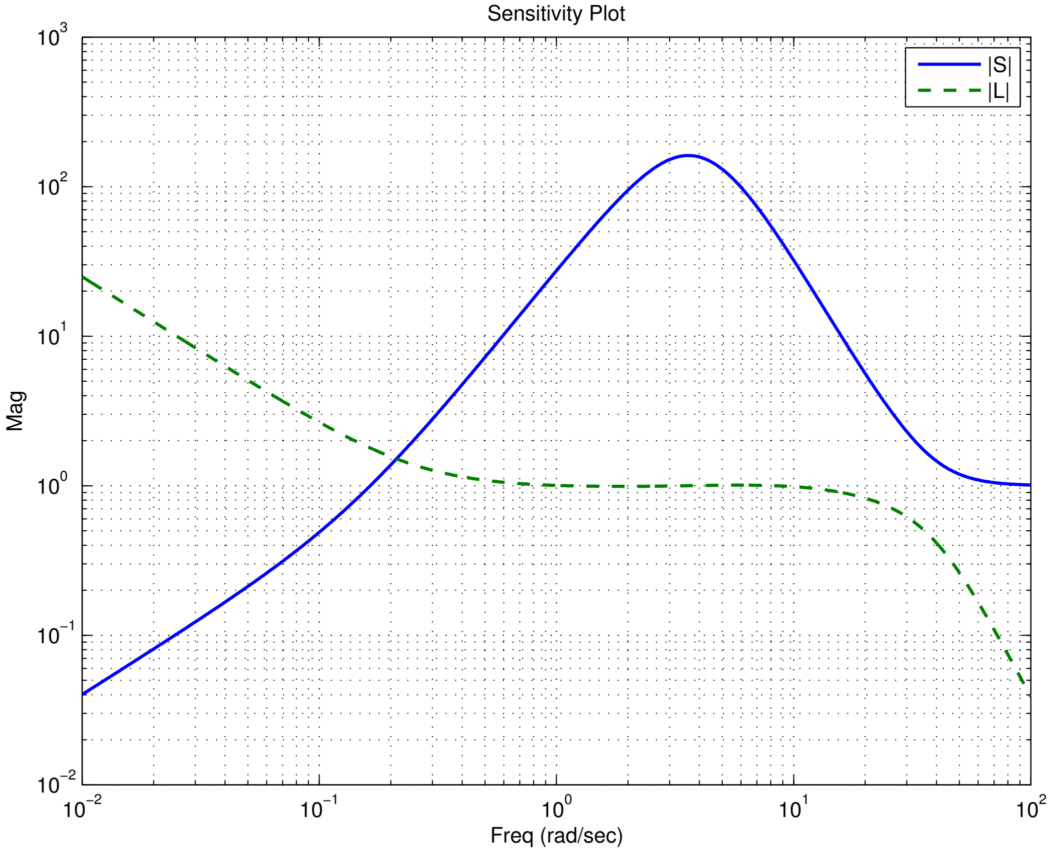


Figure 5: Sensitivity plot of the cart problem.

Difficulty in this example is that the open-loop system is unstable, so $L(\mathbf{j}\omega)$ must encircle the critical point $\Rightarrow$ hard for $L(\mathbf{j}\omega)$ to get too far away from the critical point.

Summary

- LQG gives you a great way to design a controller for the nominal system.

- But there are no guarantees about the stability/performance if the actual system is slightly different.
  - Basic analysis tool is the **Sensitivity Plot**

- No obvious ways to tailor the specification of the LQG controller to improve any lack of robustness
  - Apart from the obvious "lower the controller bandwidth" approach.
  - And sometimes you need the bandwidth just to stabilize the system.

- Very hard to include additional robustness constraints into LQG
  - See my Ph.D. thesis in 1992.

- Other tools have been developed that allow you to **directly** shape the sensitivity plot $|S(\mathbf{j}\omega)|$
  - Called $\mathcal{H}_\infty$ and μ

- **Good news:** Lack of robustness is something you should look for, but it is not always an issue.

Cite as: Jonathan How, course materials for 16.31 Feedback Control Systems, Fall 2007. MIT OpenCourseWare (http://ocw.mit.edu), Massachusetts Institute of Technology. Downloaded on [DD Month YYYY].