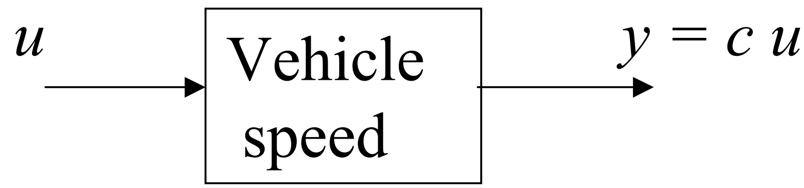
Two basic concepts of feedback systems: Performance, Robustness

- An extremely simplified model

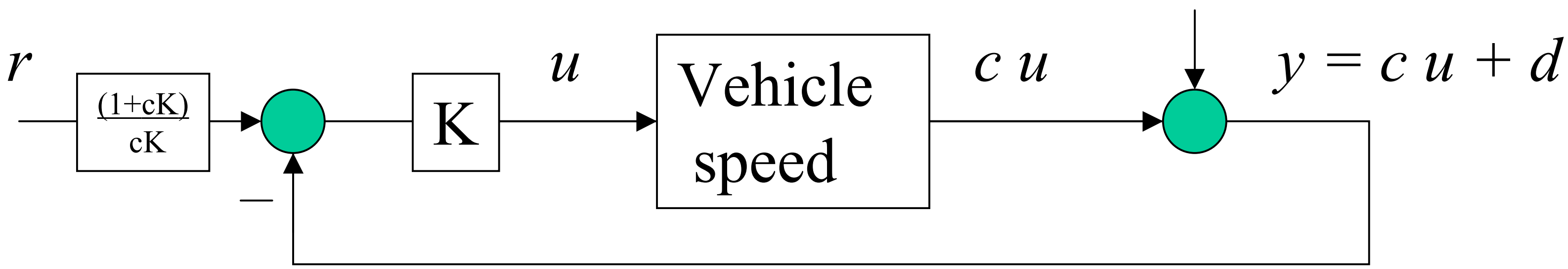


(All models are wrong, some are useful[DR]*)*

- Objective $y = r$ (e.g., c=100, r=50)
- Easy choice: $u = r/c$

Performance...

- Add a disturbance: $y = cu+d$ (say $d = -20$)
- Then $y = c(r/c)+d$ = 50-20 = 30 :(
- Measure y and close the loop:



- $u = K(r_f-y), r_f = (1+cK)/cK$ (adjusted ref.)
- Then $y = cK(r_f - y) + d = \ldots = r + d/(1+cK)$

Performance...

- Conclusion: As the controller gain (K) increases, the effect of the disturbance is reduced.

- ***Rule#1: High gain => Performance***

- Similar conclusion for variations in c (a bit more messy)
- So, take K = 1000000…
- Too good, too easy? …

Robustness...

- Suppose there is an unmodeled small delay between application of input and output:

  $$y(t) = cu(t-\tau)$$

- Without the disturbance,

  $$y(t) = cK[r_f(t-\tau) - y(t-\tau)]$$

- Look at the sample times τ, 2τ, 3τ, …

  $$y[n\tau] = -cK\, y[(n-1)\tau] + cKr_f$$

- If $cK > 1$, $|y(n\tau)|$ grows exponentially! :(

Robustness...

- For a continuous time version of this argument, use a Pade approximation of the delay:

$$e^{-s\tau} \cong \frac{1 - s\,\tau/2}{1 + s\,\tau/2}$$

$$y = c\frac{1 - s\,\tau/2}{1 + s\,\tau/2} K(y - r_f) \Rightarrow y = \frac{cK\left(1 - s\,\tau/2\right)}{(1 - cK)\,\tau/2\, s + (1 + cK)} r_f$$

$$poles: \quad \frac{1 + cK}{1 - cK}\frac{2}{\tau}$$

- Hence, the closed loop is unstable for *cK>1*.

Robustness...

- Conclusion: High controller gains can cause instability due to unmodeled dynamics.
- Alternative interpretation: Our controller did not respect the limitations of the model

  (Models have limitations, stupidity does not![AAR, MA])

- ***Rule#2: Respect the uncertain[GS]***

- NOTE: For unstable systems, low controller gains can lead to instability as well, but this is predicted by the model.

Robust controller design

- A nominal model by itself is not very useful. We also need an uncertainty estimate.
- Multiplicative uncertainty (relative error)

$$G(s) = [I + \Delta_m(s)]G_0(s)$$

$$|\Delta_m(j\omega)| = \frac{|G(j\omega) - G_0(j\omega)|}{|G_0(j\omega)|} \approx \frac{|y_0(j\omega) - y(j\omega)|}{|y_0(j\omega)|}$$

- Estimate the bound analytically or from data.
- *Small Gain Theorem:* the controller should satisfy

$$|T(j\omega)| < 1/|\Delta_m(j\omega)|$$

where $T = G_0C/(1+G_0C)$ (complementary sensitivity)

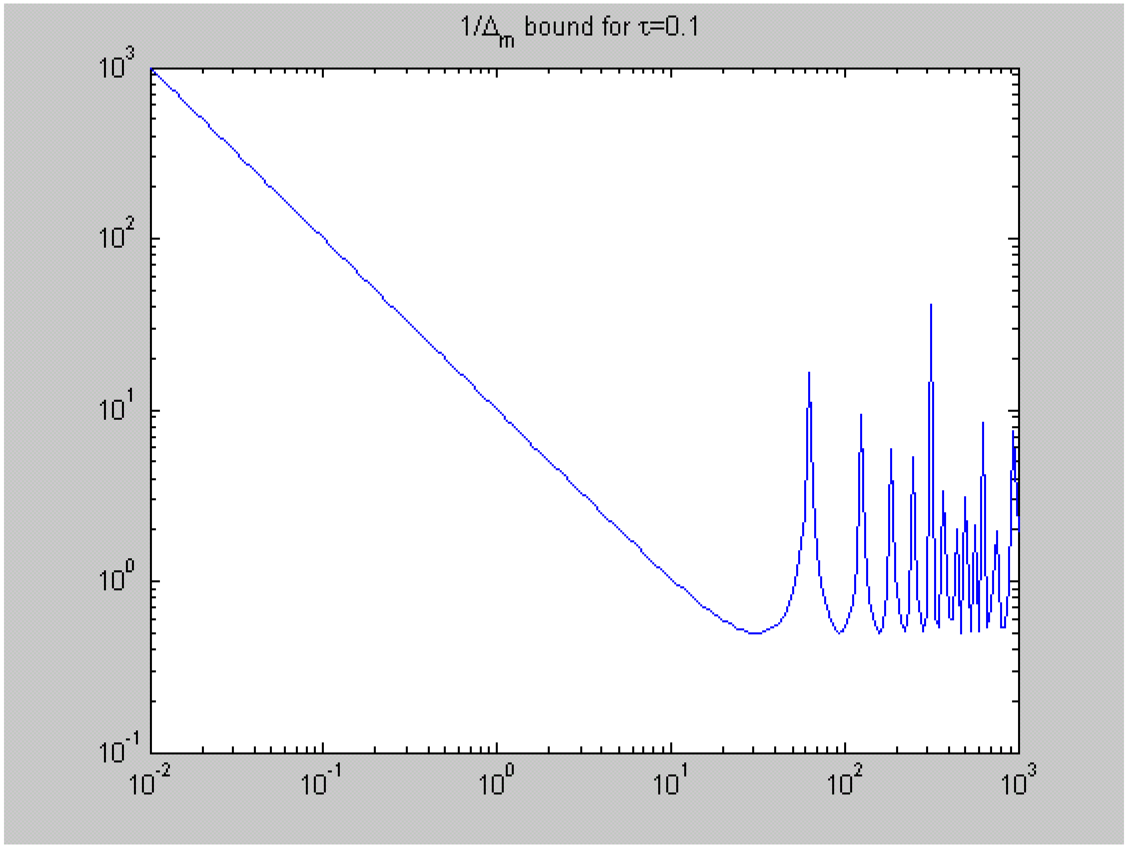
Robust controller design

- In our case, G_0=c, C=K, so T = cK/(1+cK)

T is a constant, so we must have T< 1/2 for all ω.

=> cK<1

For the plot,
>> w=logspace(-2,3,200);
>> d=exp(j*w*.1);
>> loglog(w,1./abs(1-d))



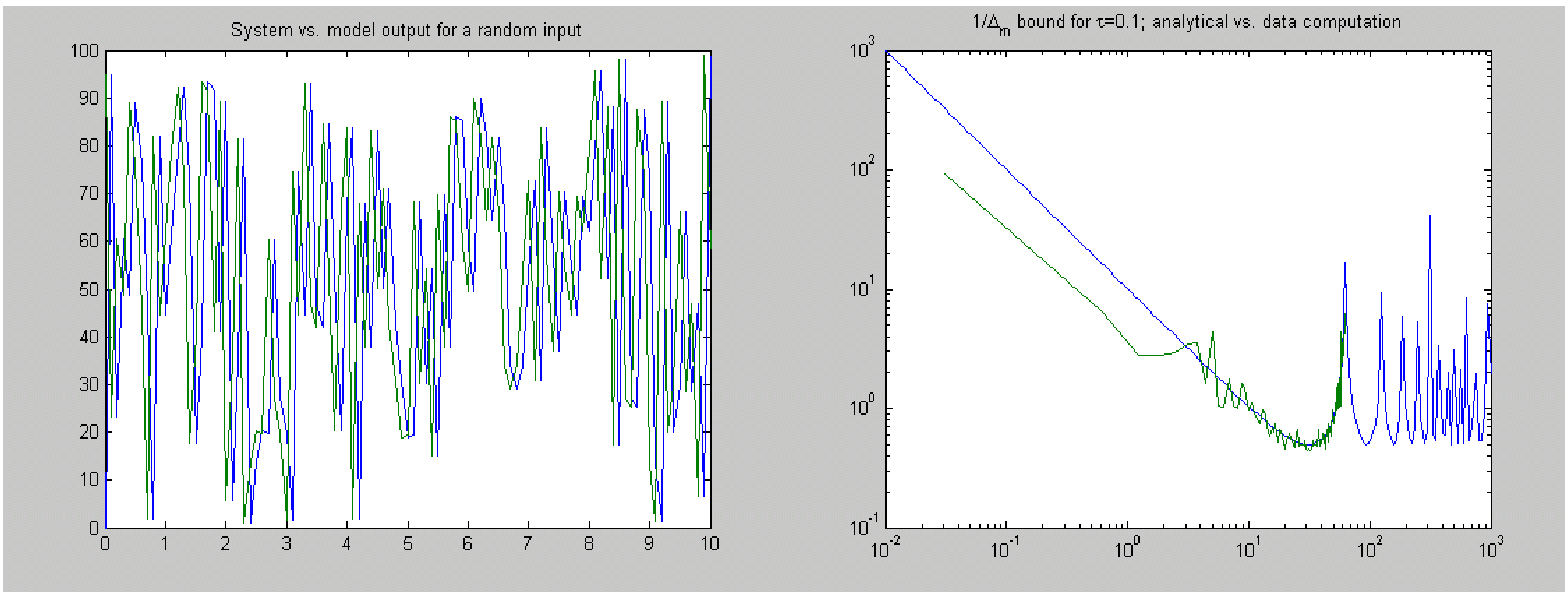
Robust controller design

- Estimating the uncertainty from data:
  - Collect data with a random input

>> dd=abs(fft(y-y0)./fft(y0));

>> ww=([1:pts]'-1)*2*pi/pts/dt;ww(1)=ww(2)/20;

- %fft frequencies for pts number of points, sampled every dt.
- %w(1)=w(2)/20, an arbitrary choice.



Robust controller design

- Dynamic controllers: Getting better performance at some frequencies.
- The uncertainty constraint is not active at low frequencies => we should be able to increase the gain there.
- PI control: $u = (K_p s + K_i)/s[e]$, (e = error = r-y)
  - For simplicity, consider the I-part only; for our easy system we do not need the P- nor the D-part. We loose some performance at high frequencies but our typical d is a low frequency signal.
  - Controller gain = $|C(j\omega)| = K_i/\omega$; large at low frequencies. Also interpreted as an internal model of constant disturbances.

Robust controller design

- Find the complementary sensitivity (nominal)

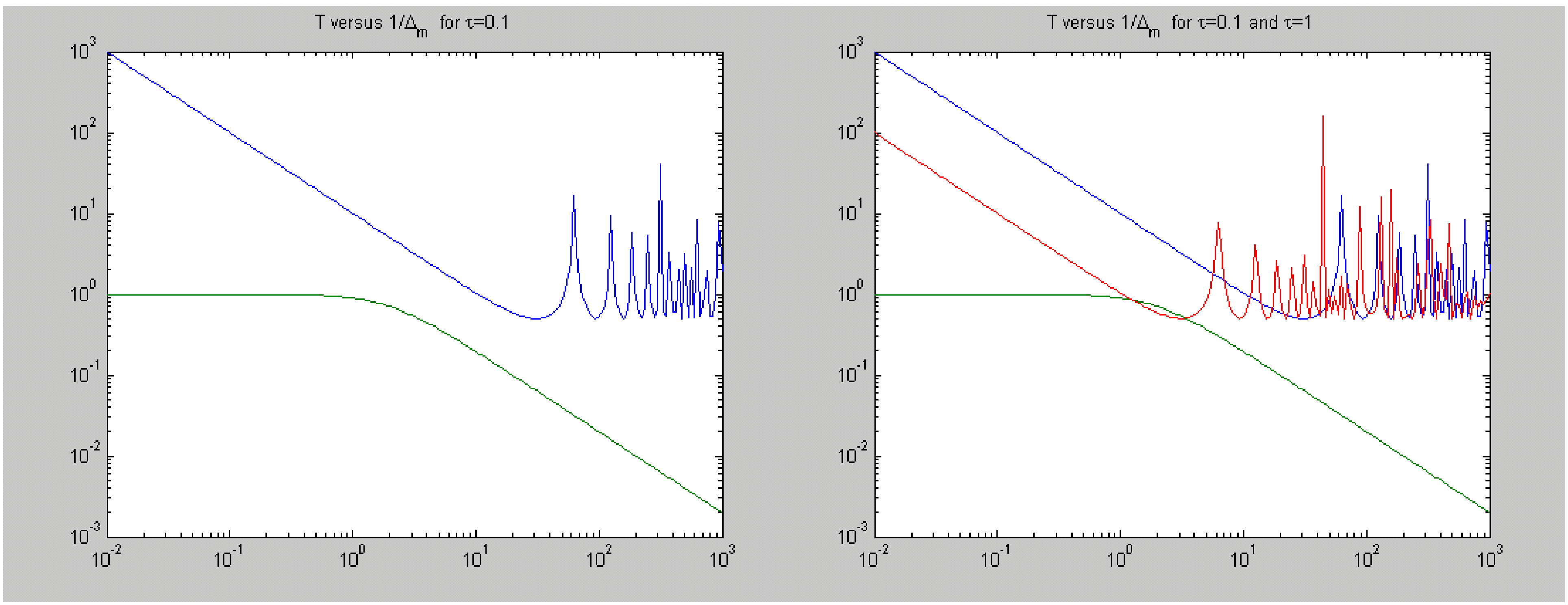
$$T(s) = \frac{cK_i/s}{1 + cK_i/s} = \frac{cK_i}{s + cK_i}$$

- Should have $|T(j\omega)| < 1/|\Delta_m(j\omega)|$
- Roughly, closed-loop bandwidth = $cK_i < \omega_{max}$ (uncertainty crossover frequency), which is ~ 10.
- Leave some margin too, so $K_i = 2/c$
- Analysis: (takes a while)

  Stable closed loop, y converges to r for any constant d, small errors for slowly varying d! :)

Robust controller design

- The better performance has a price: This controller cannot tolerate arbitrary delays…
  - Left figure: closed-loop T vs. uncertainty bound;
  - Right figure: what happens if the delay increases…



Conclusions

- Concepts demonstrated by this example:
  - Performance-robustness tradeoff
    - Improvement with dynamic controllers
  - Our expectations from the model
    - *All models are wrong, some are useful*[DR]
  - Model uncertainty restricts our expectations from the controller
    - *Controllers have limitations, stupidity does not!*[AAR, MA]

Conclusions

- Concepts carefully hidden in this example:
  - High order systems can impose further constraints on what is achievable
    - Additional constraints can come from non-invertible elements (RHP poles-zeros)
    - More of the same from nonlinear elements
  - Nominal controller design can get quite complicated for high order systems
    - and so does the analysis (though the principles stay largely the same)