Multivariable Control Design

NASA Workshop: Winter 2025

Lecture 9: Robust Synthesis With DK Iteration

Key Takeaways

This lecture focuses on designing a controller to optimize the closed-loop performance with an uncertain plant.

This is formulated as a μ −synthesis problem and is solved via DK-iteration alternating between two steps:

1. D-step: Solve for μ upper bound (D-scales) on closed-loop robust performance for a fixed controller.
2. K-step: Solve for optimal controller with fixed D-scales.

Several examples are given to illustrate this method.

Robust Control Problem Formulation

Robust Control Problem Formulation

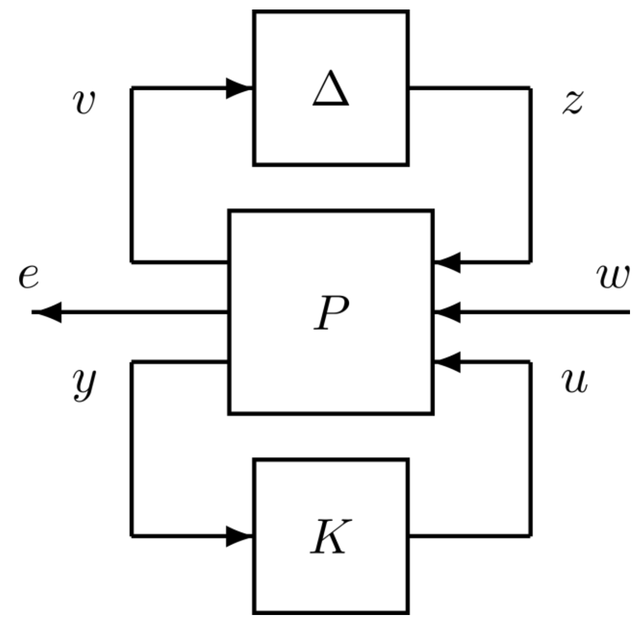
Plant: *P* is the open-loop interconnection containing nominal plant model, performance and uncertainty weighting functions

Three inputs: uncertainty perturbations *w*, disturbances *d*, and controls *u*

Three outputs: uncertainty perturbations *v*, errors *e* and measurements *y*

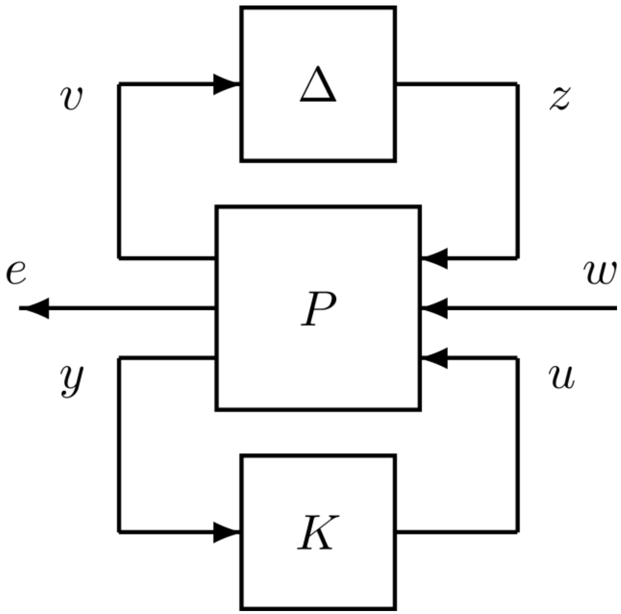
Uncertainty: $\Delta \in \mathbf{\Delta}$ where $\mathbf{\Delta}$ is a (block-structured) set of dynamic and parametric uncertainties.

Controller: *K* is the controller that uses measurements *y* to compute the control inputs *u*.



Robust Control Problem Formulation

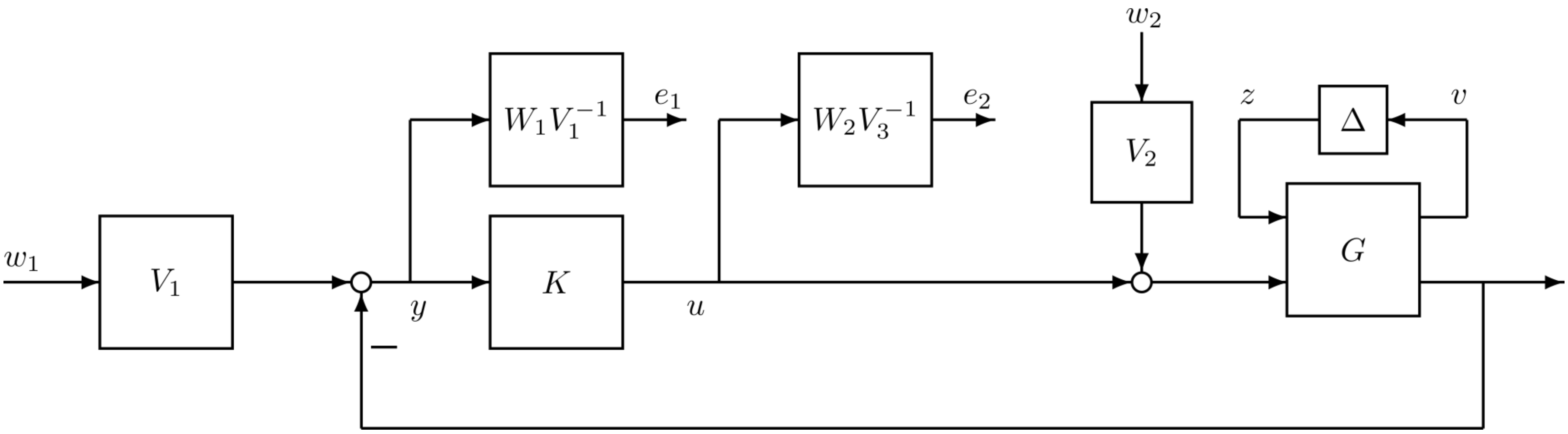
Goal: Design *K* to optimize the closed-loop performance objectives in the presence of the assumed model uncertainty.



Robust Control Problem Formulation

Goal: Design *K* to optimize the closed-loop performance objectives in the presence of the assumed model uncertainty.

Example: Design *K* to make the weighted closed-loop from w to e have small gain in the presence of uncertainty.

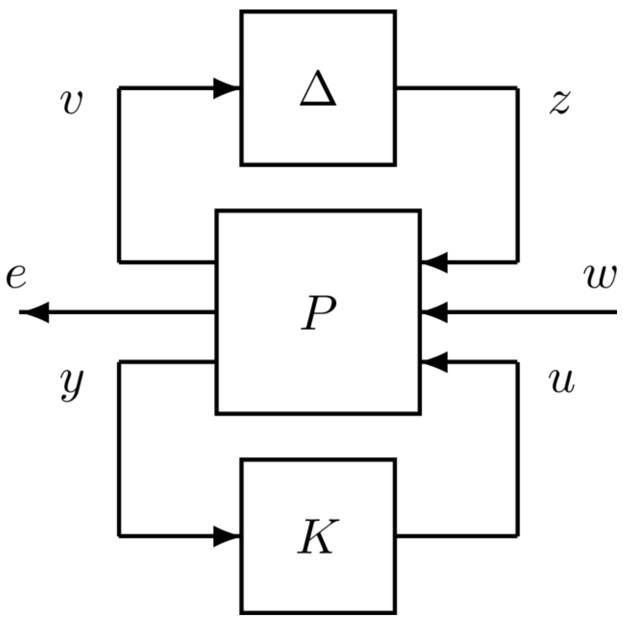


Robust Control Problem Formulation

To make this more precise, we consider the following set of uncertain systems:

$$\mathcal{M} := \{F_U(P, \Delta) \; : \; \Delta \in \mathbf{\Delta}, \; \|\Delta\|_\infty \leq 1\}$$

The goal is to design a controller *K* such that $F_L(\tilde{P}, K)$ is stable and achieves performance $||F_L(\tilde{P}, K)||_\infty < 1$ for all plants $\tilde{P} \in \mathcal{M}$, i.e., achieves robust stability and robust performance.



Robust Performance

(Revisiting Lecture 7)

MIMO Small Gain Condition

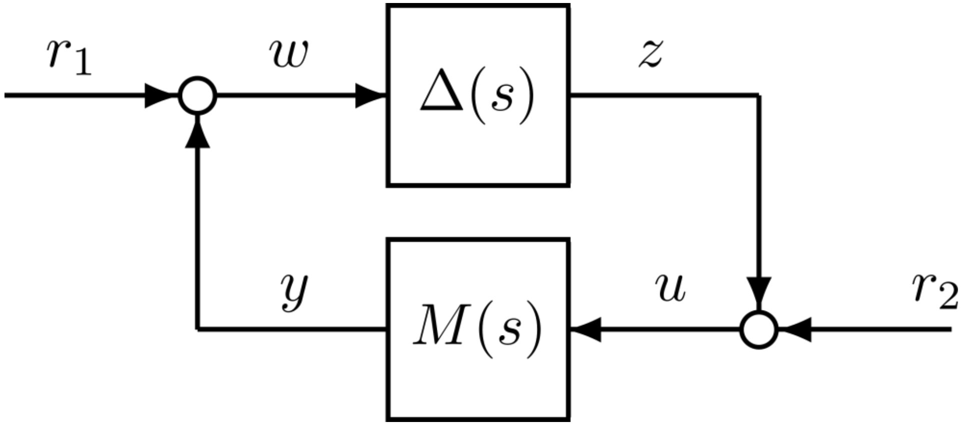
Theorem: Consider the positive feedback system below where $M(s)$ is stable.

A) If $||\mathrm{M}||_\infty \leq 1$ then the feedback system is stable for all $\Delta(s)$ that are stable and is norm-bounded $||\Delta||_\infty < 1$.

B) If $||\mathrm{M}||_\infty > 1$ then there is a stable $\Delta(s)$ with $||\Delta||_\infty < 1$ such that the feedback system is unstable.

We used this to prove robust stability based on $||\mathrm{M}||_\infty$.

We can use this in reverse: Prove a bound on $||\mathrm{M}||_\infty$ by checking for robust stability.

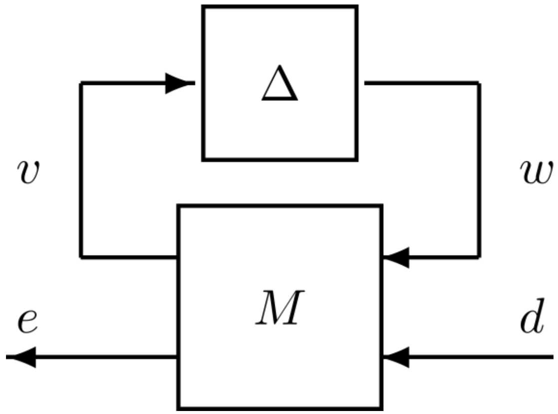


Robust Performance

Assume M and Δ are both stable. Also assume $\|M_{11}\|_\infty < 1$ so that the LFT is robustly stable.

What condition on M ensures robust performance?

$$\sup_{\|\Delta\|_\infty \leq 1} \|F_U(M,\Delta)\|_\infty < 1$$



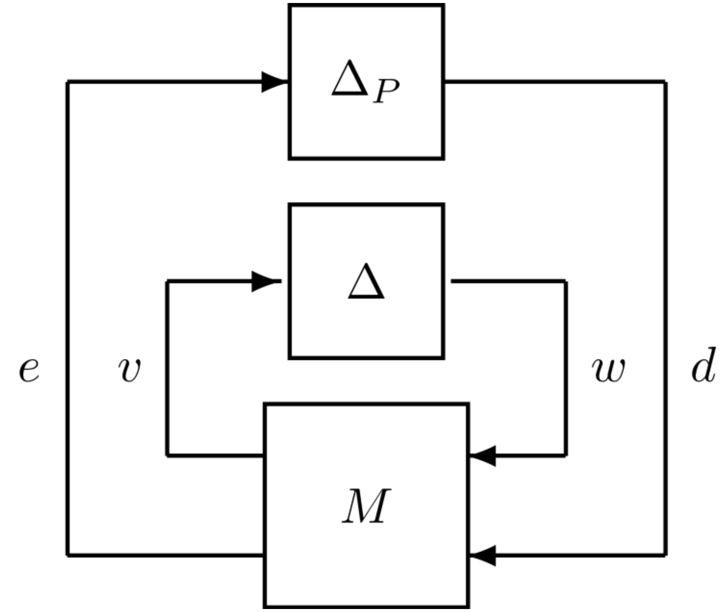
Robust Performance

Assume M and Δ are both stable. Also assume $\|M_{11}\|_\infty < 1$ so that the LFT is robustly stable.

What condition on M ensures robust performance?

$$\sup_{\|\Delta\|_\infty \leq 1} \|F_U(M, \Delta)\|_\infty < 1$$

Robust performance is equivalent to robust stability of M with an additional unstructured uncertainty block Δ_P on the performance channel.



Robust Performance

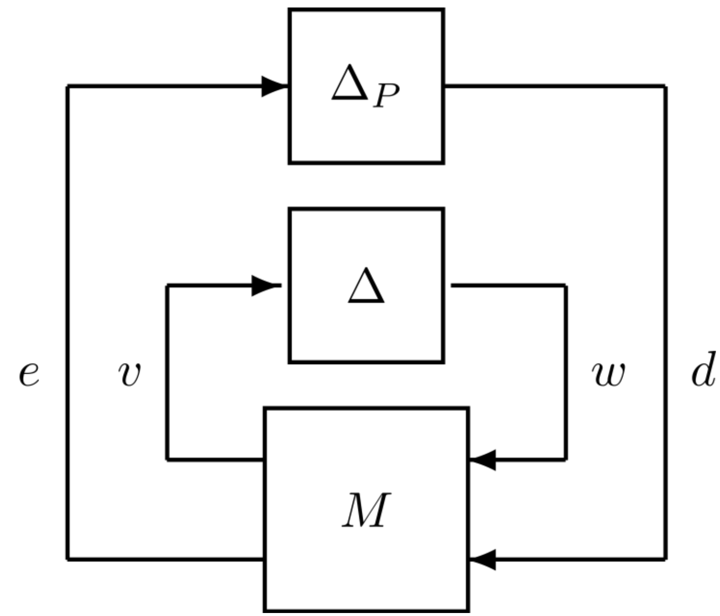
Assume M and Δ are both stable. Also assume $\|M_{11}\|_\infty < 1$ so that the LFT is robustly stable.

What condition on *M* ensures robust performance?

$$\sup_{\|\Delta\|_\infty \leq 1} \|F_U(M,\Delta)\|_\infty < 1$$

We analyze robust performance for the original problem by checking robust stability of *M* with respect to the larger, augmented uncertainty:

$$\left\| \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_P \end{bmatrix} \right\|_\infty \leq 1$$



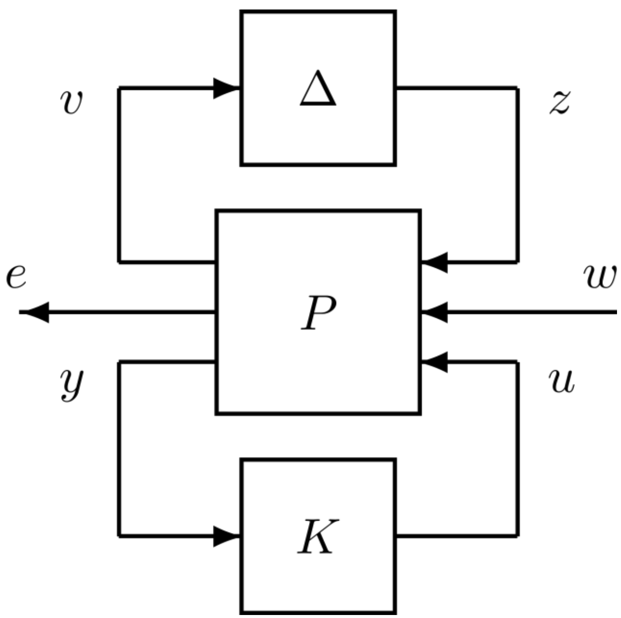
DK-Iteration For Robust Synthesis

Robust Control Problem Formulation

To make this more precise, we consider the following set of uncertain systems:

$$\mathcal{M} := \{F_U(P,\Delta) \,:\, \Delta \in \mathbf{\Delta},\, \|\Delta\|_\infty \leq 1\}$$

The goal is to design a controller *K* such that $F_L(\tilde{P}, K)$ is stable and achieves performance $||F_L(\tilde{P}, K)||_\infty < 1$ for all plants $\tilde{P} \in \mathcal{M}$, i.e., achieves robust stability and robust performance.



Robust Control Problem Formulation

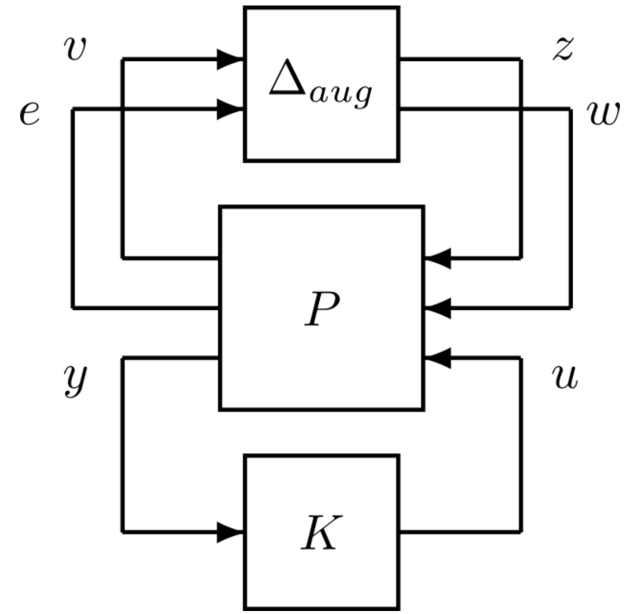
Create an augmented uncertainty with an additional "performance" block:

$$\mathbf{\Delta}_{aug} := \left\{ \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_P \end{bmatrix} : \Delta \in \mathbf{\Delta},\ \Delta_P \in \mathbb{C}^{n_w \times n_e} \right\}$$

The corresponding model set is:

$$\mathcal{M}_{aug} := \{ F_U(P, \Delta_{aug}) \, : \, \Delta_{aug} \in \mathbf{\Delta}_{aug},\ \|\Delta_{aug}\|_\infty \leq 1 \}$$

The goal is to design a controller K such that $F_L(\tilde{P}, K)$ is stable for all plants $\tilde{P} \in \mathcal{M}_{\text{aug}}$, i.e., achieves robust stability for the augmented uncertainty.

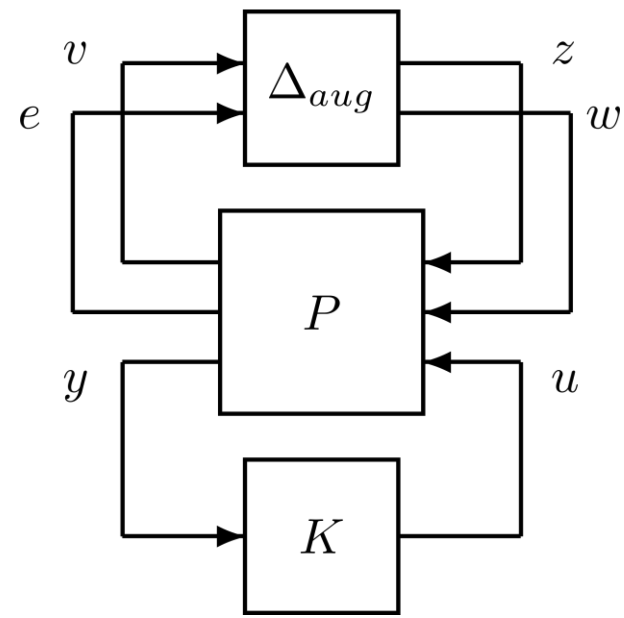


Small Gain Condition

The augmented model set is:

$$\mathcal{M}_{aug} := \{F_U(P, \Delta_{aug}) \,:\, \Delta_{aug} \in \boldsymbol{\Delta}_{aug},\ \|\Delta_{aug}\|_\infty \le 1\}$$

$F_L(\tilde{P}, K)$ is stable for all plants $\tilde{P} \in \mathcal{M}_{\text{aug}}$ if and only if



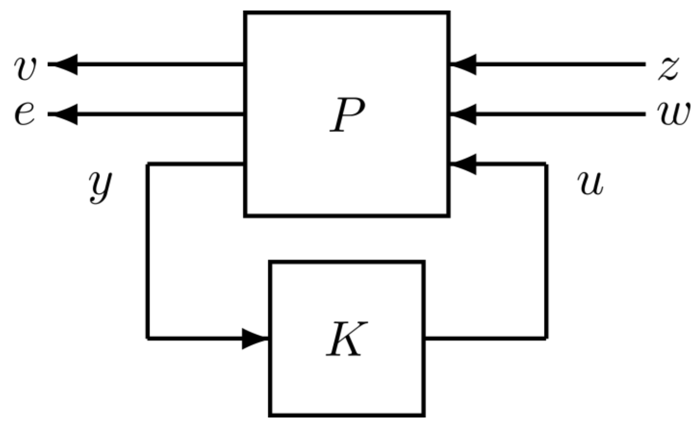
Small Gain Condition

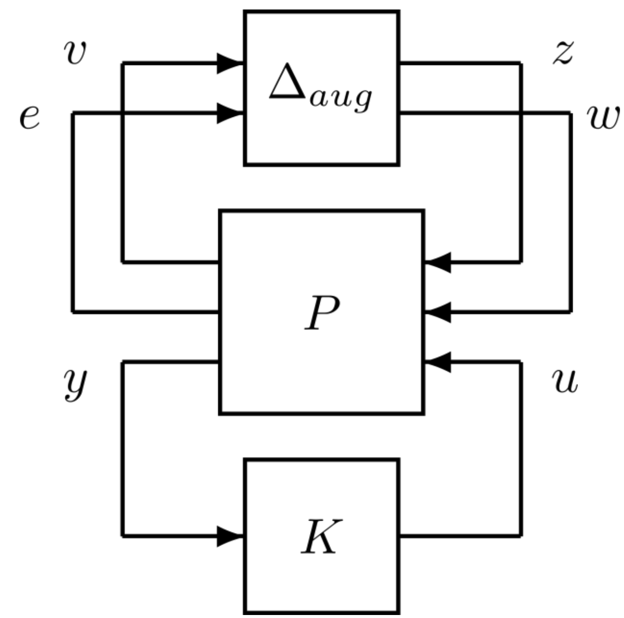
The augmented model set is:

$$\mathcal{M}_{aug} := \{F_U(P, \Delta_{aug}) \, : \, \Delta_{aug} \in \mathbf{\Delta}_{aug}, \, \|\Delta_{aug}\|_\infty \leq 1\}$$

$F_L(\tilde{P}, K)$ is stable for all plants $\tilde{P} \in \mathcal{M}_{\text{aug}}$ if and only if

- $F_L(P, K)$ is stable, and
- $\max_{\omega} \mu_{\Delta_{aug}}(\, F_L(P, K)(j\omega)\,) < 1$

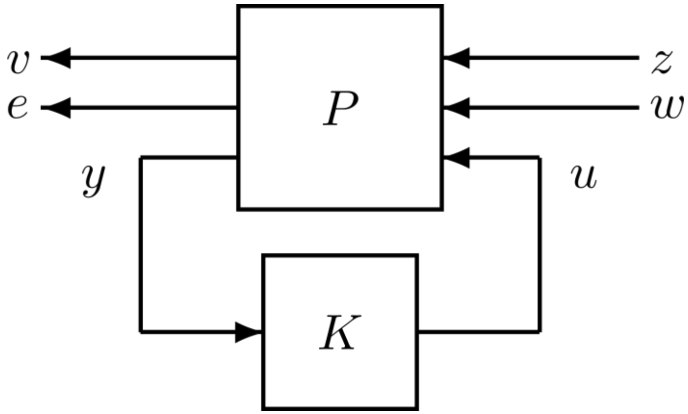




μ-Synthesis

Design K to stabilize the nominal plant P and minimize the peak value of $\mu_{\Delta_{aug}}$ of the closed-loop $F_L(P,K)$:

$$\min_{K \text{ stabilizing}} \left[\max_{\omega} \mu_{\Delta_{aug}}\left(F_L(P,K)(j\omega) \right) \right]$$



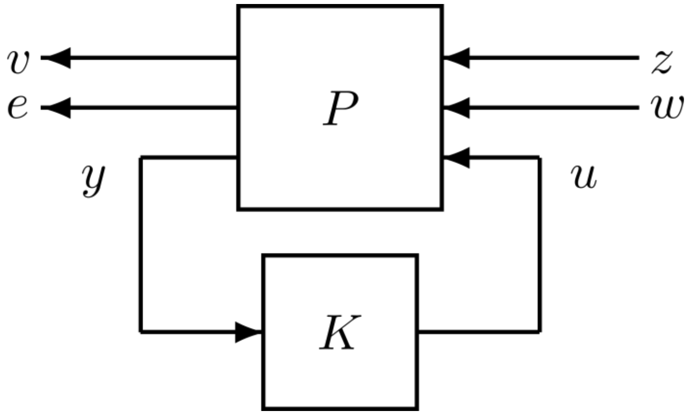
μ-Synthesis

Design K to stabilize the nominal plant P and minimize the peak value of $\mu_{\Delta_{aug}}$ of the closed-loop $F_L(P,K)$:

$$\min_{K \text{ stabilizing}} \left[\max_{\omega} \mu_{\Delta_{aug}}(\, F_L(P,K)(j\omega)\,) \right]$$

Note: This contains H_∞ as a special case:

- If there is no uncertainty (*v/z* channels are not present) then $\mu_{\Delta_{aug}}(\, F_L(P,K)(j\omega)\,) = \bar{\sigma}(\, F_L(P,K)(j\omega)\,)$.
- The problem simplifies to $\min_{K \text{ stabilizing}} ||F_L(P,K)||_\infty$



μ-Synthesis via Upper Bound

It is difficult to exactly compute μ. Hence, we replace μ by its upper bound for tractability:

$$\mu_{\boldsymbol{\Delta}}(M) \leq \inf_{\mathrm{D} \in \boldsymbol{D}_{\boldsymbol{\Delta}}} \bar{\sigma}(DMD^{-1})$$

where $\boldsymbol{D}_{\boldsymbol{\Delta}}$ is the set of *D*-scales. These satisfy $D\Delta = \Delta\, D$ for every $D \in \boldsymbol{D}_{\boldsymbol{\Delta}}$ and $\Delta \in \boldsymbol{\Delta}$. In most problems the upper bound is "close" to the true value of μ.

The design problem becomes:

$$\min_{K \text{ stabilizing}} \left[\max_{\omega} \min_{D(j\omega) \in \boldsymbol{D}_{\boldsymbol{\Delta}}} \bar{\sigma}\left(D(j\omega)\, F_L(P,K)(j\omega)\, D(j\omega)^{-1}\right) \right]$$

μ-Synthesis via Upper Bound

The design problem becomes:

$$\min_{K \text{ stabilizing}} \left[\max_{\omega} \min_{D(j\omega) \in \boldsymbol{D}_{\boldsymbol{\Delta}}} \bar{\sigma}\left(D(j\omega)\, F_L(P,K)(j\omega)\, D(j\omega)^{-1}\right) \right]$$

$D(j\omega)$ is chosen independently at each frequency so we switch the order of the inner max/min:

$$\min_{K \text{ stabilizing}} \min_{D \in \boldsymbol{D}_{\boldsymbol{\Delta}}} || D\ F_L(P,K)\ \ D^{-1} ||_{\infty}$$

Here *D* is an LTI system with $D(j\omega) \in \boldsymbol{D}_{\boldsymbol{\Delta}}$.

It can be shown (with some technical details) that we can restrict *D* to be stable and minimum phase.

- The max singular value is invariant to multiplication by unitary matrices on the left and/or right.
- We can use this to modify the phase of the blocks of $D(j\omega)$.

μ-Synthesis via Upper Bound

The design problem is:

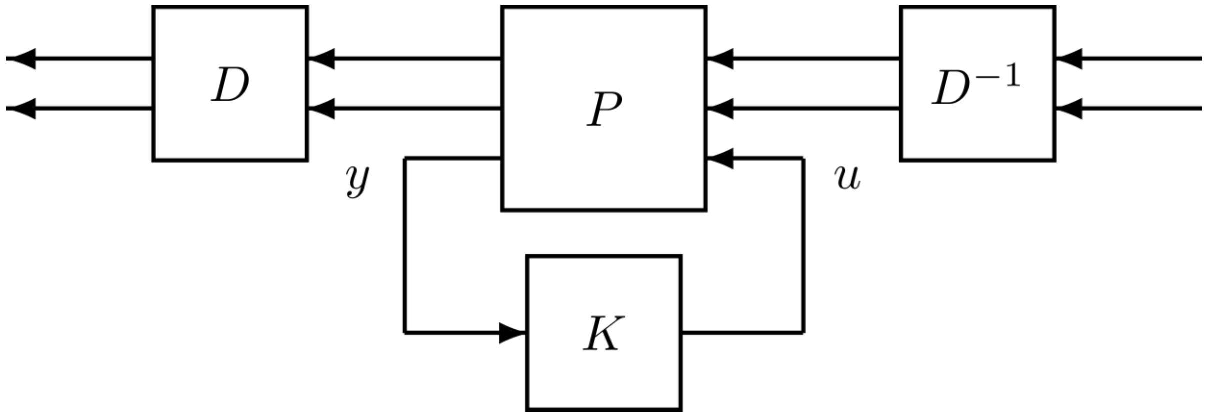
$$\min_{K \text{ stabilizing}} \min_{D \in \boldsymbol{D}_{\boldsymbol{\Delta}}} || D\, F_L(P, K)\, D^{-1} ||_\infty$$

This is a non-convex problem, in general.

This optimization is typically "solved" (approximately) by an iterative approach, referred to as D – K iteration:

D-Step: Hold *K* fixed and minimize over *D*.

K-Step: Hold *D* fixed and minimize over *K*.

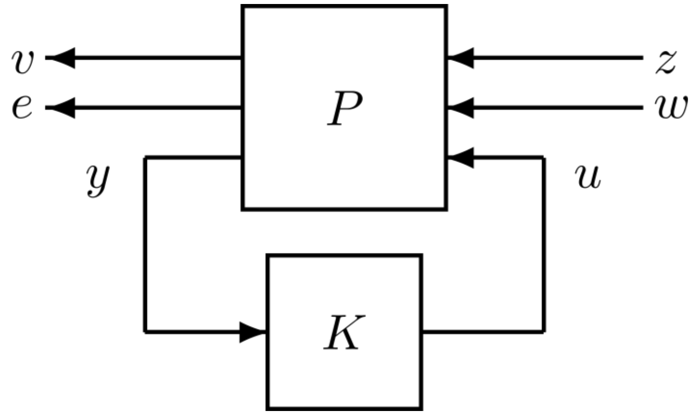


D-K Iteration: K-Step

1. Initialize D=I and solve:

$$\min_{K \text{ stabilizing}} \| F_L(P, K) \|_\infty$$

This is a standard H_∞ problem and yields K_1.



D-K Iteration: D-Step

2. Hold $K = K_1$ fixed and solve:

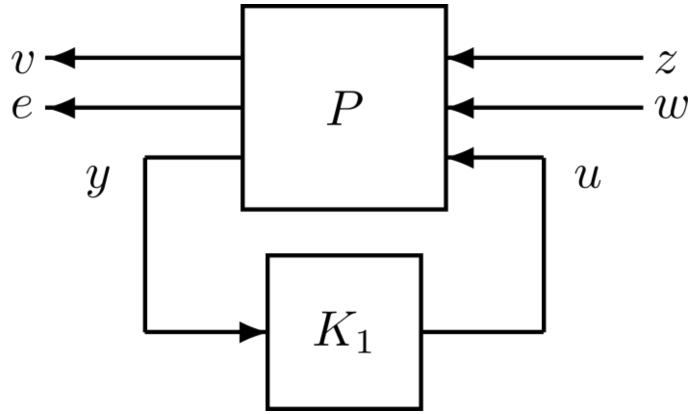
$$\min_{D \in \boldsymbol{D}_{\boldsymbol{\Delta}}} || D\, F_L(P, K_1)\, D^{-1} ||_\infty$$

This is a μ upper bound problem and is solved by:

a. Solve the μ upper bound problem on a set of freqs. $\{\omega_k\}_{k=1}^{N}$:

$$\min_{D(\omega_k) \in \boldsymbol{D}_{\boldsymbol{\Delta}}} \bar{\sigma}(D(j\omega_k)\, F_L(P, K_1)(j\omega_k)\, D(j\omega_k)^{-1})$$

b. Fit $\{D(j\omega_k)\}_{k=1}^{N}$ with a stable, minimum phase, LTI system $D_2(s)$ of a given state order. This is done with freq. domain fitting (`fitmagfrd`).



D-K Iteration: K-Step

3. Hold $D = D_2$ fixed and solve:

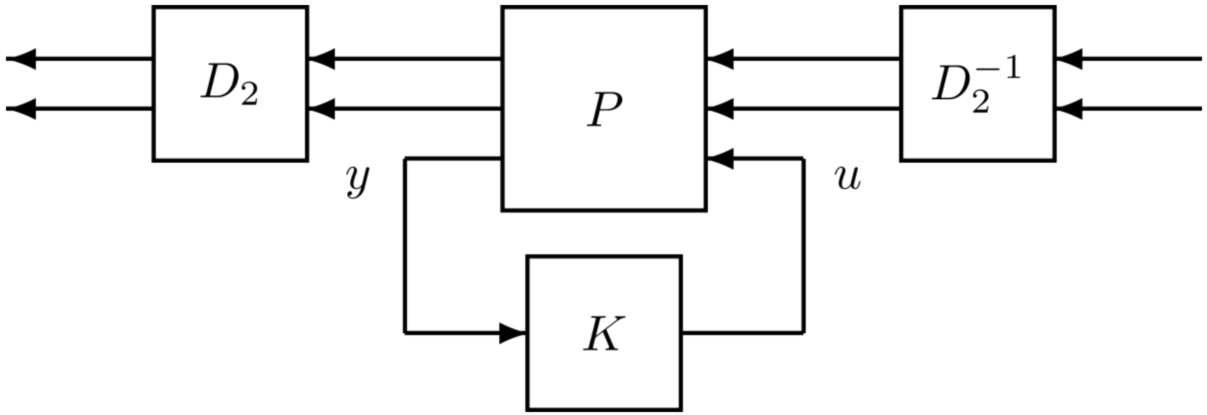
$$\min_{K \text{ stabilizing}} || D_2 \, F_L(P, K) \, D_2^{-1} ||_\infty$$

This is equivalent to an H_∞ problem on a scaled plant:

$$\min_{K \text{ stabilizing}} || F_L\left(\hat{P}, K\right) ||_\infty$$

where the scaled plant is:

$$\hat{P} := \begin{bmatrix} D_2 & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} D_2^{-1} & 0 \\ 0 & I \end{bmatrix}$$



D-K Iteration: K-Step

3. Hold $D = D_2$ fixed and solve:

$$\min_{K \text{ stabilizing}} || D_2 \, F_L(P, K) \, D_2^{-1} ||_\infty$$

This is equivalent to an H_∞ problem on a scaled plant:

$$\min_{K \text{ stabilizing}} || F_L\left(\hat{P}, K\right) ||_\infty$$

where the scaled plant is:

$$\hat{P} := \begin{bmatrix} D_2 & 0 \\ 0 & I \end{bmatrix} P \begin{bmatrix} D_2^{-1} & 0 \\ 0 & I \end{bmatrix}$$

This *K*-step yields K_2 and the iteration proceeds back to the *D*-step. This continues for a set number of iteration or until no further improvement is obtained.

D-K Iteration: Shortcomings

- We approximated μ by its upper-bound. This is not a serious problem since the value of μ and its upper-bound are often close.

D-K Iteration: Shortcomings

- We approximated μ by its upper-bound. This is not a serious problem since the value of μ and its upper-bound are often close.
- We solved for the D-scale by optimizing on a finite frequency grid and fitting with a given order. This is only a mild restriction since we can choose the frequency grid and order to approximate any rational functions.

D-K Iteration: Shortcomings

- We approximated μ by its upper-bound. This is not a serious problem since the value of μ and its upper-bound are often close.
- We solved for the D-scale by optimizing on a finite frequency grid and fitting with a given order. This is only a mild restriction since we can choose the frequency grid and order to approximate any rational functions.
- We used a coordinate-wise minimization over *(D,K)*. The D − K iteration is not guaranteed to converge to a global, or even local minimum. This is the most serious limitation of the procedure.

D-K Iteration: Shortcomings

- We approximated μ by its upper-bound. This is not a serious problem since the value of μ and its upper-bound are often close.
- We solved for the D-scale by optimizing on a finite frequency grid and fitting with a given order. This is only a mild restriction since we can choose the frequency grid and order to approximate any rational functionsl
- We used a coordinate-wise minimization over *(D,K)*. The D − K iteration is not guaranteed to converge to a global, or even local minimum. This is the most serious limitation of the procedure.

Despite these drawbacks, the D − K iteration appears to work well on many engineering problems

D-K Iteration: Practical Issues

- The fitting procedure is more complicated and leads to high-order D-scales when the uncertainty contains repeated (complex or real) blocks of the form $\delta \cdot I$
- The approach can be generalized to include additional scalings (*G*-scales) if the system includes real parametric uncertainty.

The iteration works best when the uncertainty contains is a moderate number (<10) of full blocks and a small number (<3) of repeated scalar blocks.

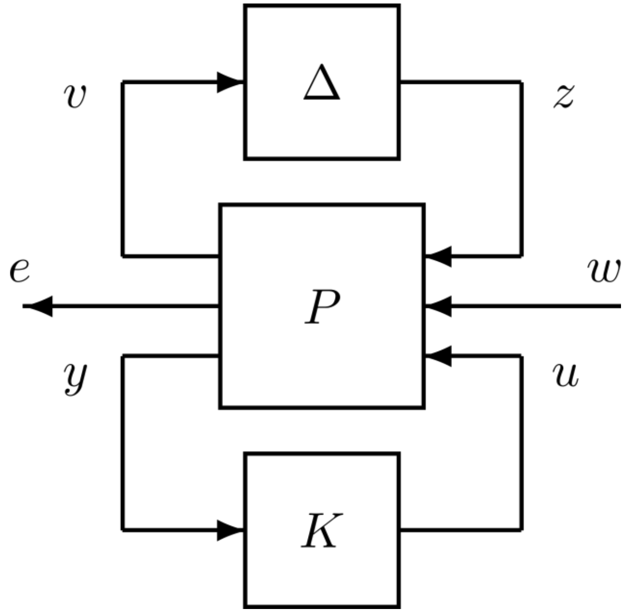
MUSYN Syntax

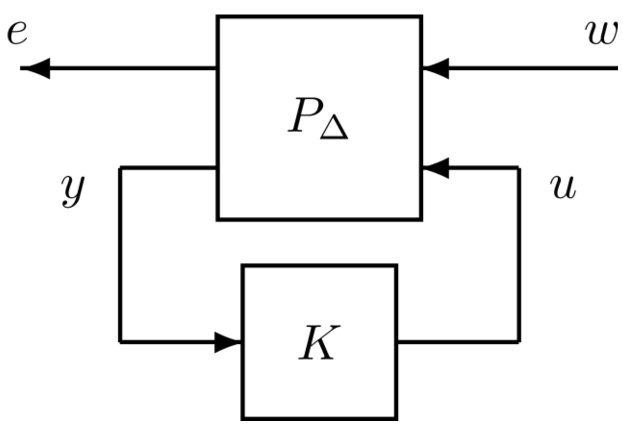
Construct uncertain plant $P_\Delta = F_U(P, \Delta)$ using uncertain objects.

[K,CLPERF,INFO] = musyn(P_Delta,Ny,Nu)

This performs the DK-iteration and returns:

- K is best controller found by the DK-iteration,
- CLPERF is the final (best) upper bound on closed-loop μ.
- INFO stores information, e.g., D-scales, from each iteration.



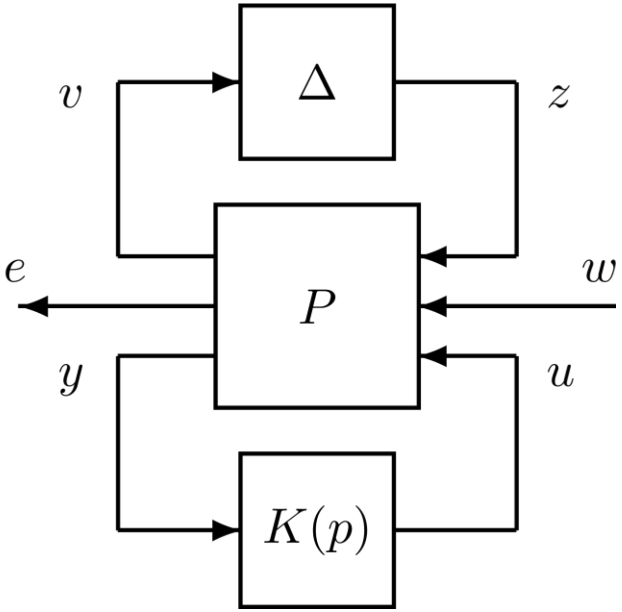


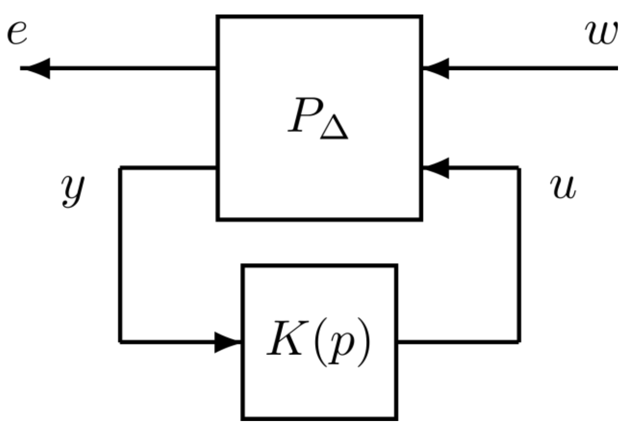
Structured MUSYN Syntax

DK-iteration can be performed to design a structured controller.

1. Construct uncertain plant $P_\Delta = F_U(P,\Delta)$ using uncertain objects.
2. Construct structured controller $K(p)$ using tunable parameters.
3. Form parameterized closed-loop $CL = F_L(P_\Delta, K(p))$.
4. [Kp,CLPERF,INFO] = musyn(CL,Ny,Nu)

This alternates between a D-step and a K-step with structured H_∞.





Example: Flexible Rocket

Flexible Rocket

Plant: 10-state model with short-period dynamics and four flexible modes.

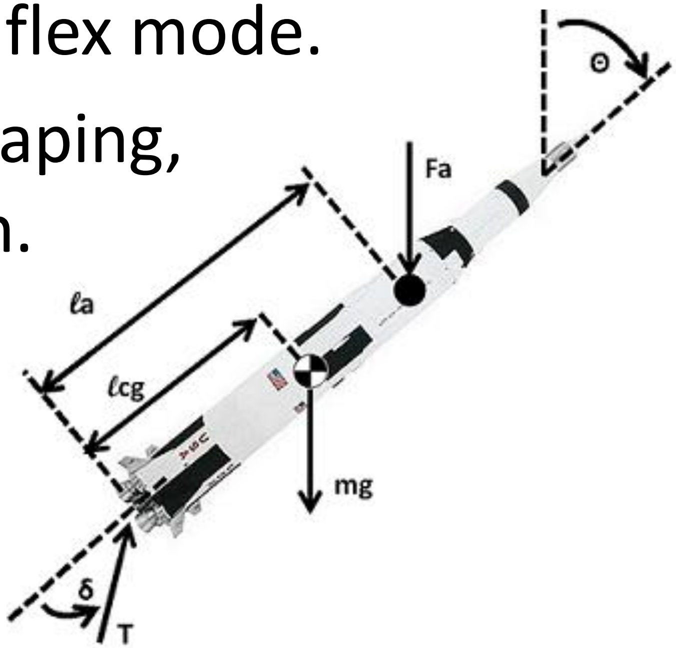
- Input: Thrust vector, u_{tvc} (units of 10^9 lbs)
- Output: Pitch rate measurement, y_{gyro} (rad/sec),

Issue: The control is difficult due to the closeness of the Unstable short-period mode and first flex mode.

Controllers: Compare classical loopshaping, H_∞ mixed sensitivity, and DK-iteration.

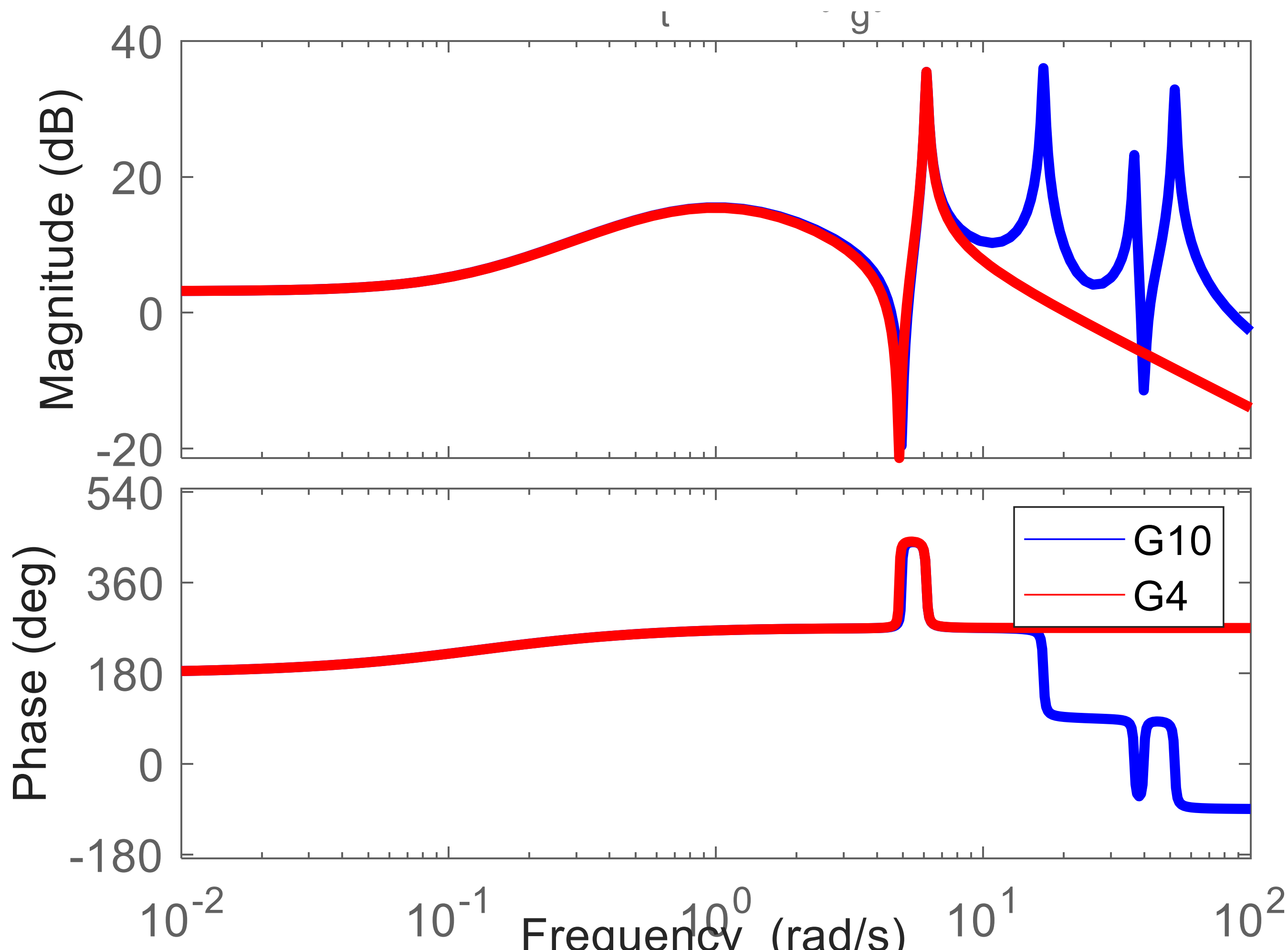
File: FlexibleRocket.m

Ref: Enns, IEEE CSM, 1991.



Full and Reduced-Order Model

Issue: Short period mode has a RHP pole at s=+1rad/s and lowest flex mode is at 6.13 rad/s.

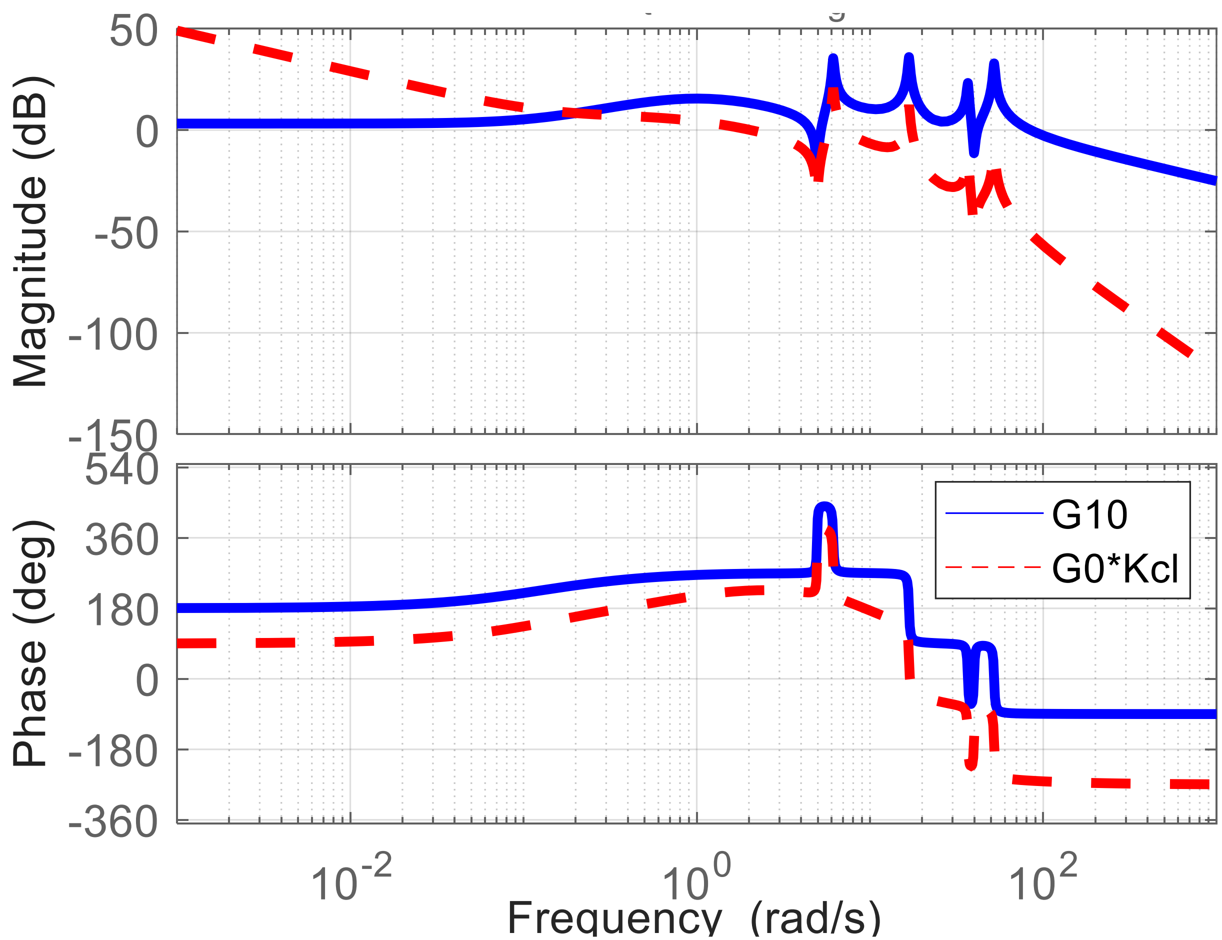


Design 1: Classical Loopshaping

Design: Loopshaping with loop bandwidth of 2rad/s using PI control with a second-order Butterworth roll-off.

Margins:

- Gain Margins = [0.44, 1.91].
- Phase margin = 46°.
- (Symmetric) disk margin = 0.31.

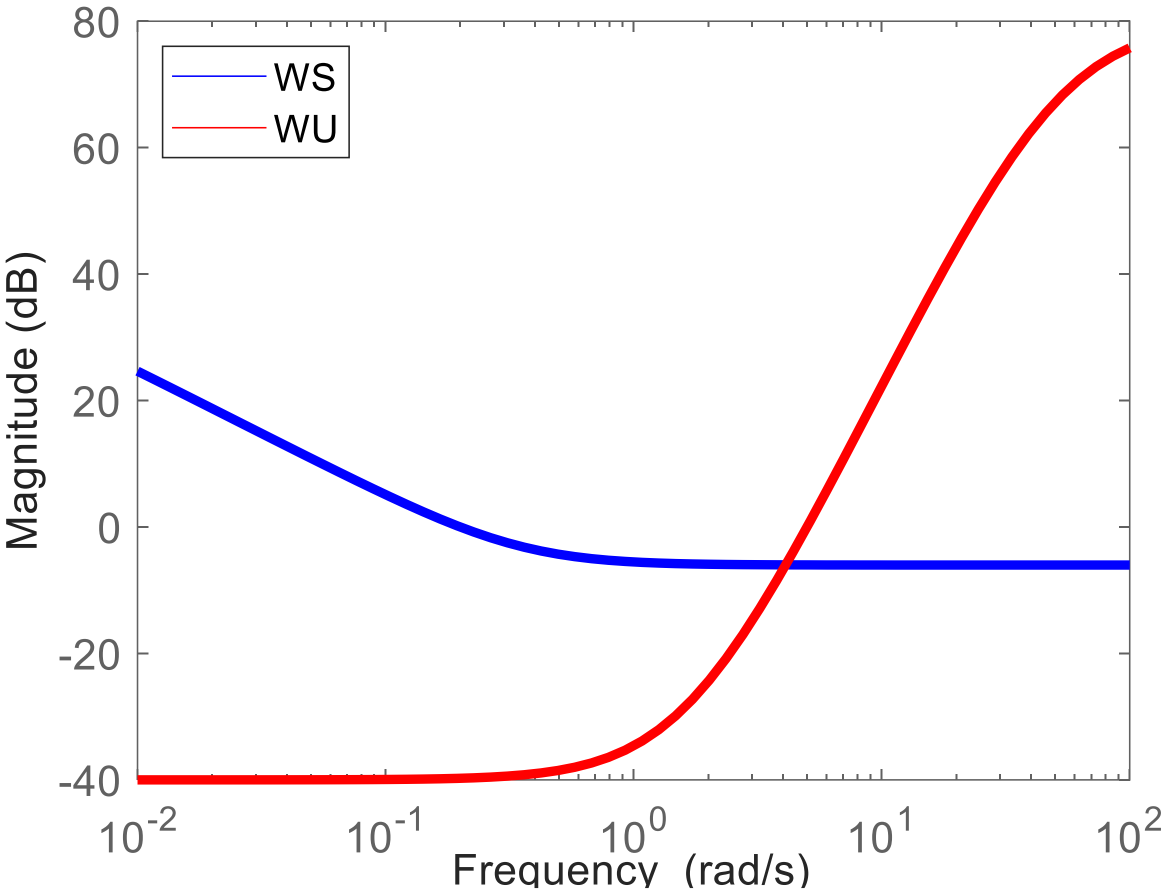


Design 2: Mixed Sensitivity Design

```
% Performance Weight
DCgain = 100;
Crossover = 0.2; HFgain = 1/2;
WS = makeweight(DCgain,Crossover,HFgain);

% Control penalty
DCgain = (0.01)^(1/4);
Crossover = 5; HFgain = 10;
WU = makeweight(DCgain,...
    Crossover,HFgain);
WU = WU^4;

% No comp sens. req.
WU = [];
```



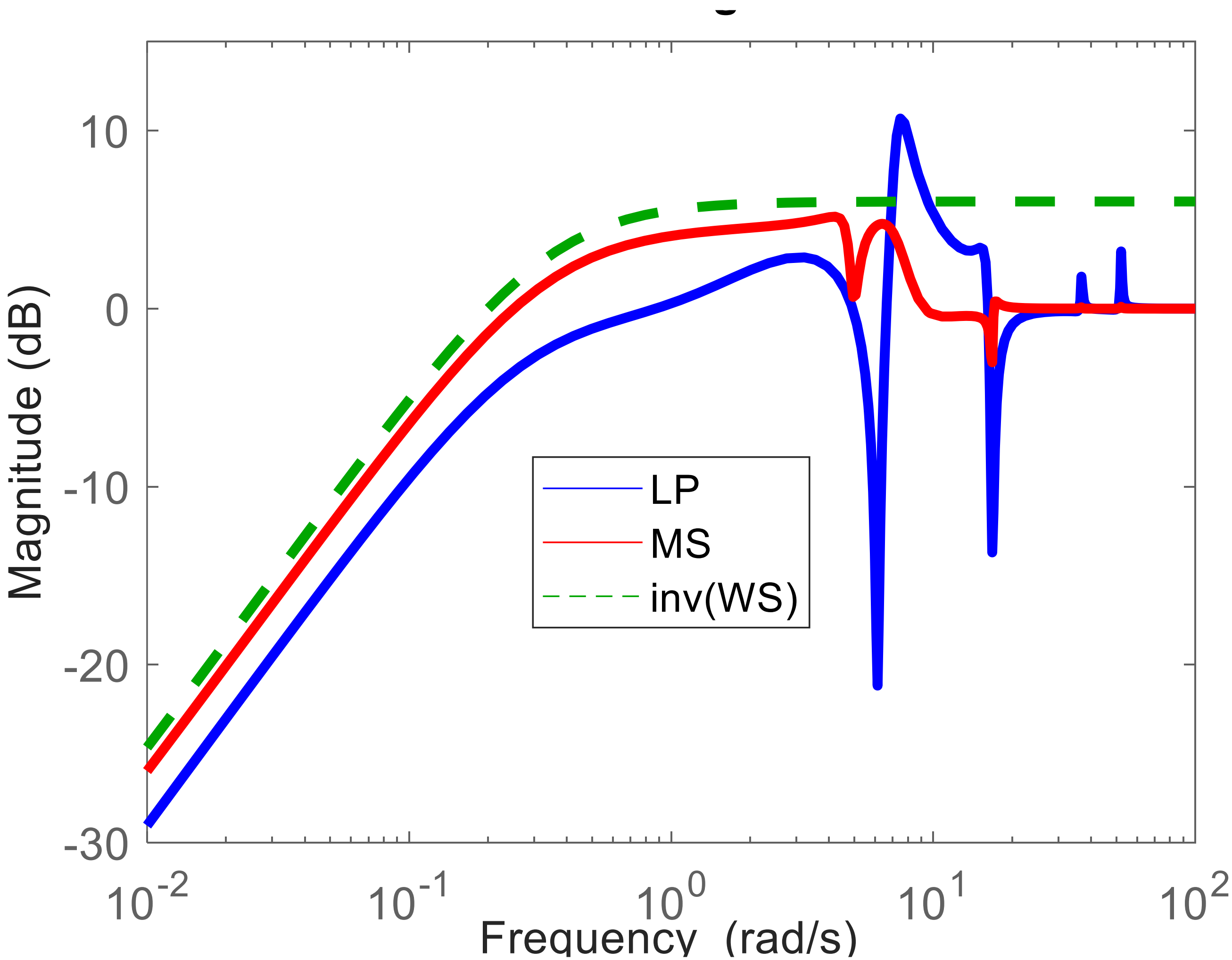
Design 2: Mixed Sensitivity Design

[Kms,clMS,gammaMS,infoMS] = mixsyn(G4,WS,WU,WT);

gammaMS = 0.8585

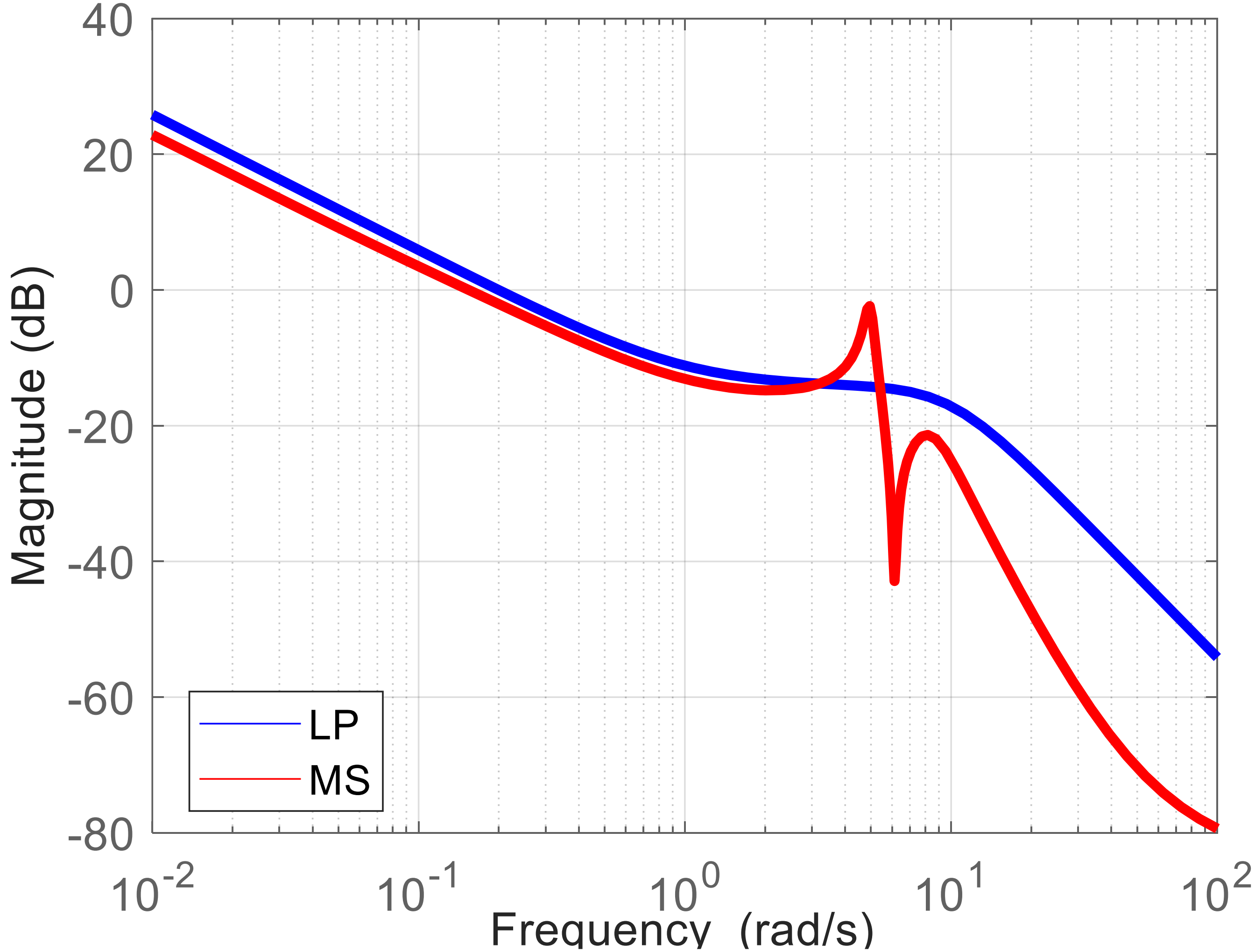
Margins:

- Gain Margins = [0.56, 2.37].
- Phase margin = 35°.
- (Symmetric) disk margin = 0.52.



Loopshaping vs. Mixed Sens. Controllers

The mixed sensitivity controller inverts the first flex mode in the plant. This will be sensitive to errors in the modal freq.



Uncertainty Modeling

The 4-state model has the short-period mode and the first flexible mode. States 1 & 3 correspond to the flex mode:

```
[A4,B4,C4,D4] = ssdata(G4);
A4([1 3],[1 3])  %  = [0    8.0000;   -4.7000   -0.1230]
```

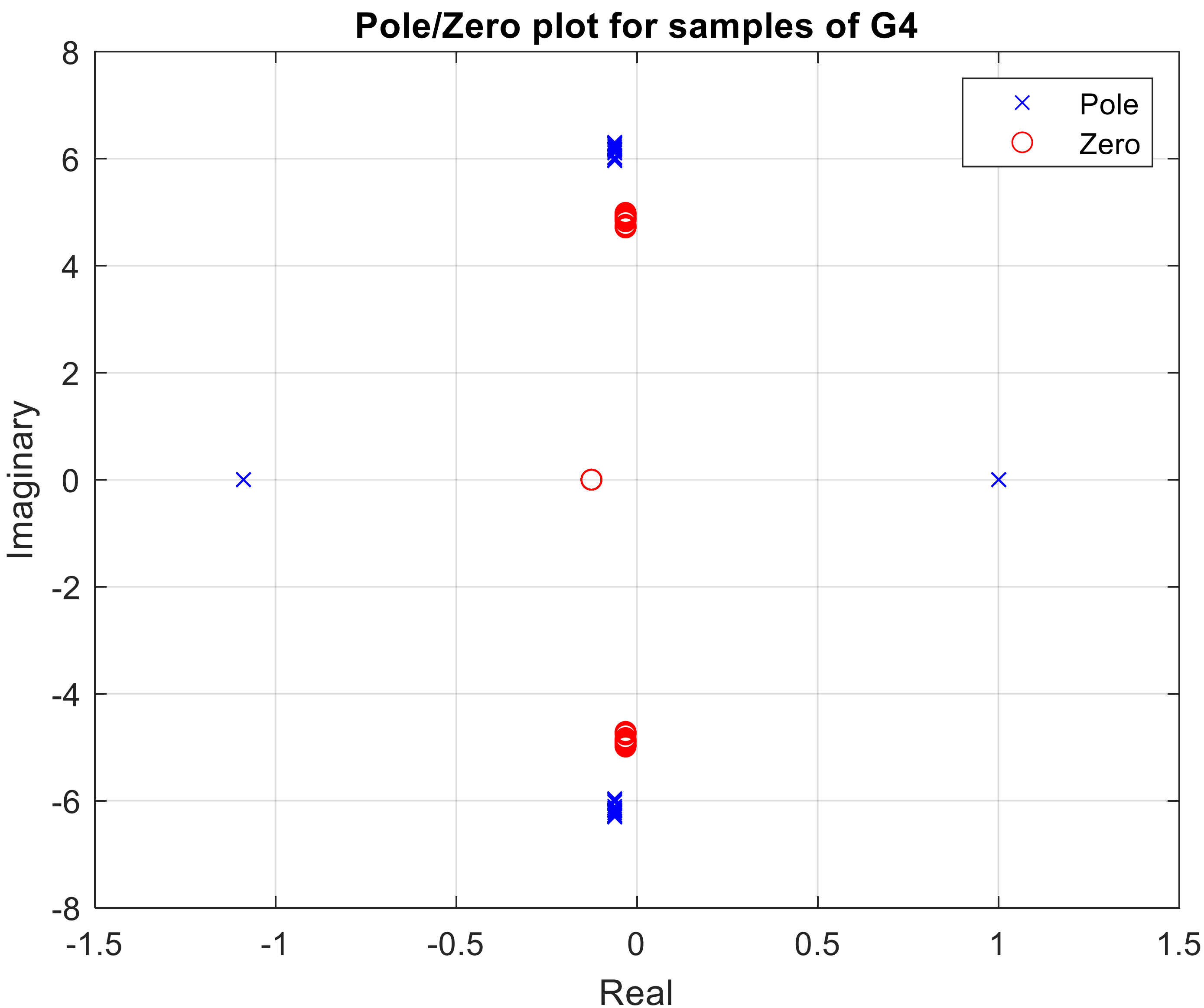
The (3,1) entry is $-\omega_{n,1}^2/8$. (Use $\hat{x}_3 = 8x_3$ to get transform to the standard second-order form $[0\ 1; -\omega_{n,1}^2\ \ -2\zeta\omega_{n,1}]$).

Introduce 6% uncertainty in this entry:

```
A4unc = umat(A4);
A4unc(3,1) = ureal('wn1sq',A4(3,1),'Percentage',[-6 6]);
```

This corresponds to $\approx 3\%$ uncertainty in $\omega_{n,1}$ and a small uncertainty in ζ.

Uncertainty Modeling: Plant Pole/Zero Plot



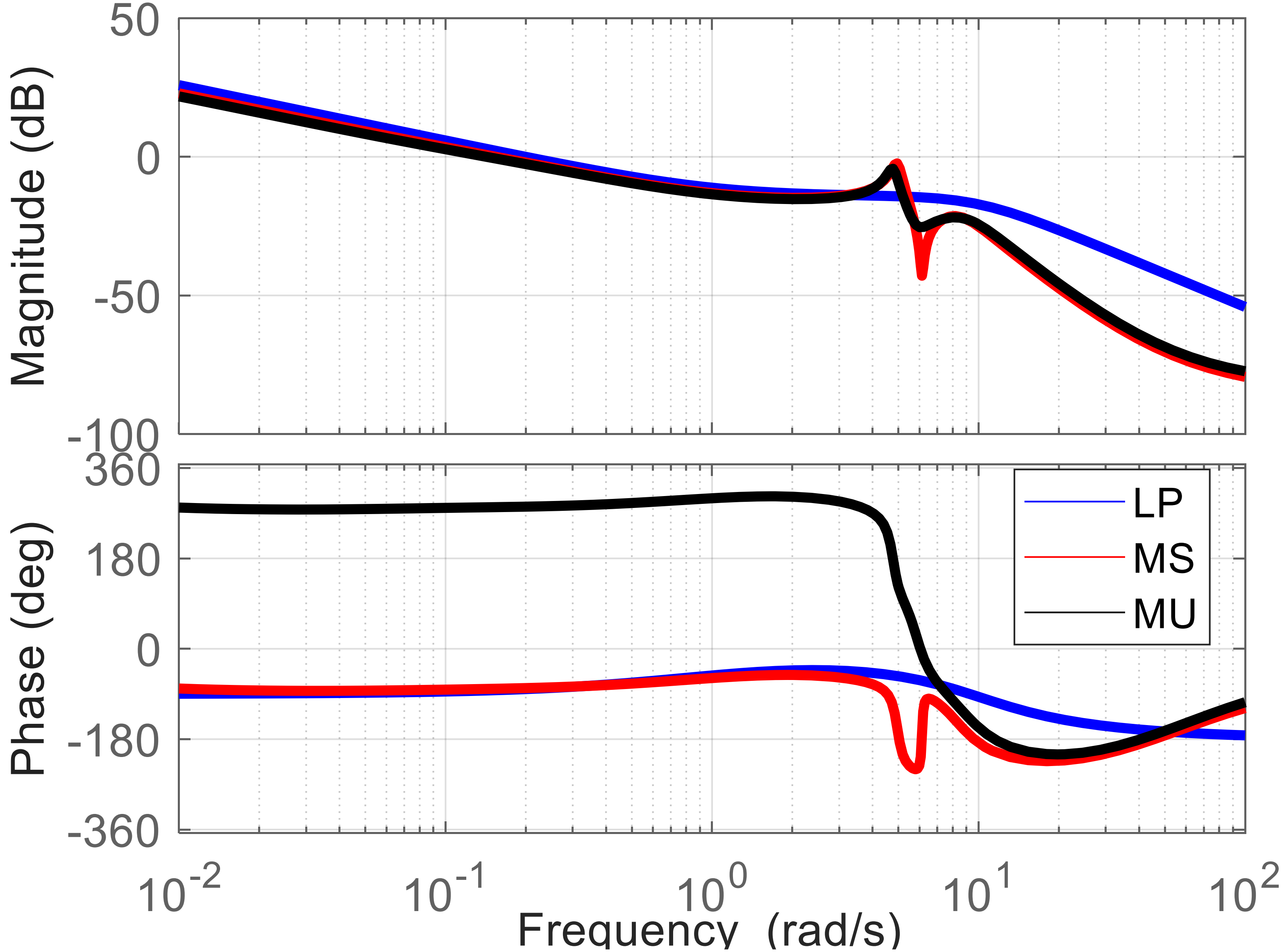
Robust Mixed Sensitivity with DK Iteration

```
% Construct design interconnection with |iconnect|.
[ny,nu] = size(G4unc);
M = iconnect; r = icsignal(ny); u = icsignal(nu); e = icsignal(ny);
M.Input = [r;u];
M.Output = [WS*e; WU*u; e];
M.Equation{1} = equate(e,r-G4unc*u);
Munc = M.System;
```

Robust Mixed Sensitivity with DK Iteration

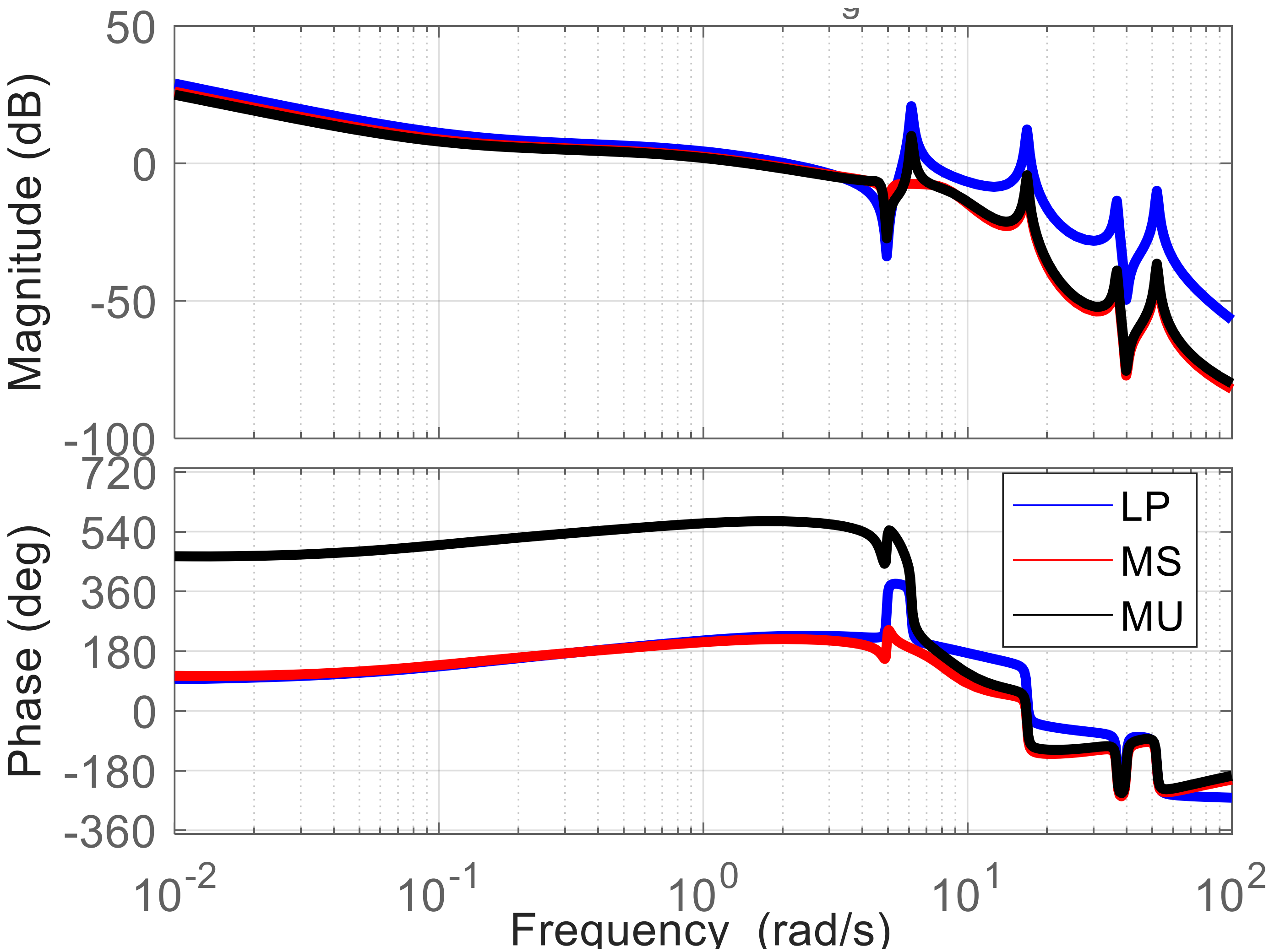
```
nmeas = 1; ncon = 1;
opt = musynOptions; opt.MaxIter = 5;
[Kmu,clpMU,infoMU] = musyn(Munc,nmeas,ncon,opt);
D-K ITERATION SUMMARY:
-----------------------------------------------------------------
              Robust performance           Fit order
-----------------------------------------------------------------
 Iter      K Step      Peak MU      D Fit         D
   1       1.107       1.107        1.119        10
   2         1         1            1.001         8
   3       0.9854      0.9854       0.9857        8
   4       0.9852      0.9852       0.9853        8
   5       0.9835      0.9835       0.9837        8
Best achieved robust performance: 0.983
```

Comparison of Controllers

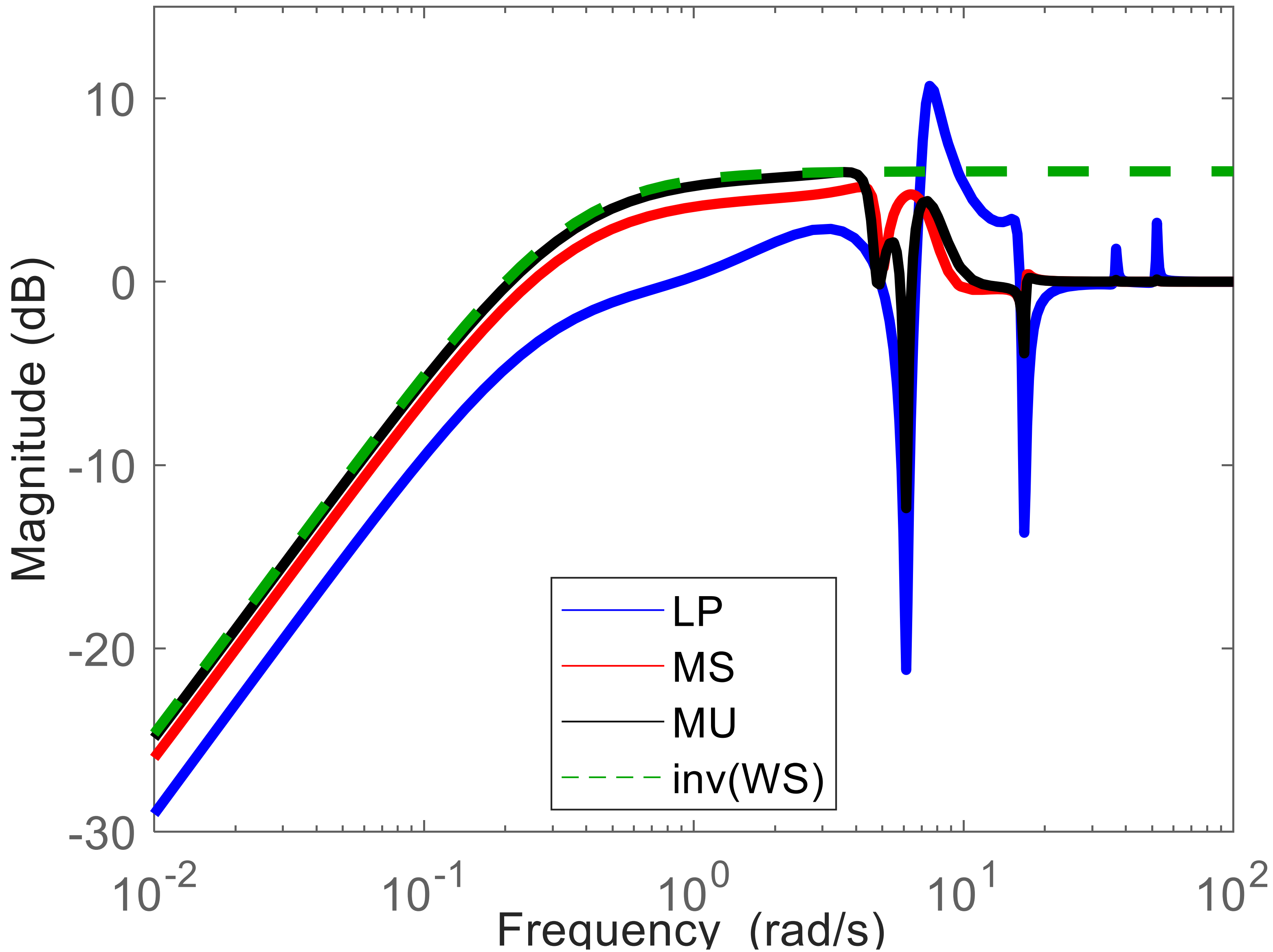


Comparison of Loops

DKSyn controller phase stabilizes the first flex mode.

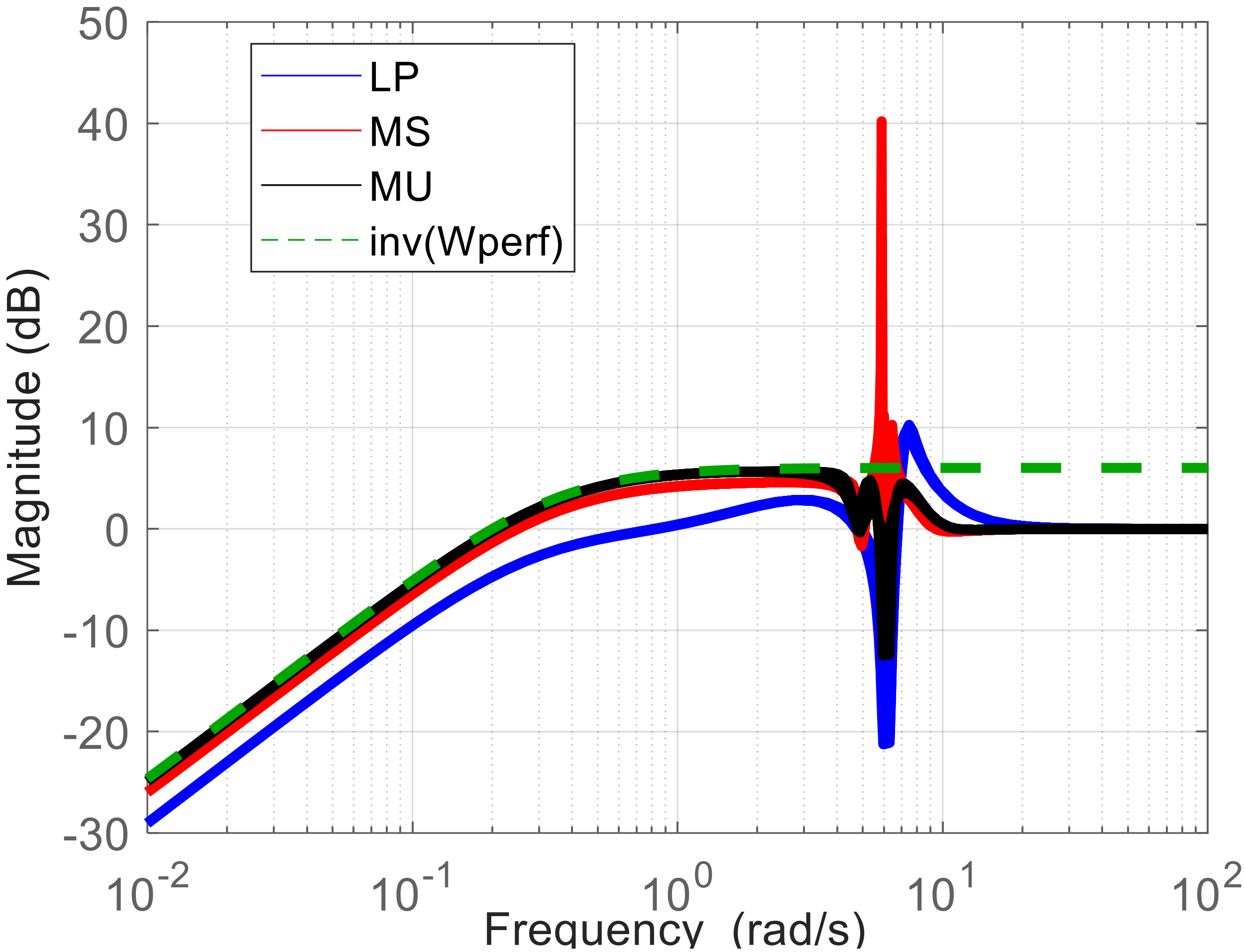


Comparison of Nominal Closed-Loop Sensitivities



Samples of Closed-Loop Sensitivity

The mixed sensitivity design is very sensitive to the uncertainty in modal frequency.



Robustness of Closed-Loops

Compute stability margins of the three closed-loops with the uncertainty in $\omega_{n,1}^2$.

```
>> SMcl = robstab(Sclunc);        % SMcl.LowerBound = 15.94
>> SMms = robstab(Smsunc);   % SMms.LowerBound = 0.76
>> SMmu = robstab(Smuunc);  % SMmu.LowerBound = 2.48
```

This margin analysis also illustrates the lack of robustness of the mixed sensitivity design to the modeled uncertainty.

The closed-loop gain for the μ controller is robust:

```
wcgMU = wcgain(Smuunc);
nominalGainMU = norm(Smuunc.Nominal,inf);
[nominalGainMU wcgMU.LowerBound wcgMU.UpperBound]
 1.9154    1.9227    1.9267
```

Example: Idealized Manufacturing Process

Example: Idealized Manufacturing Process

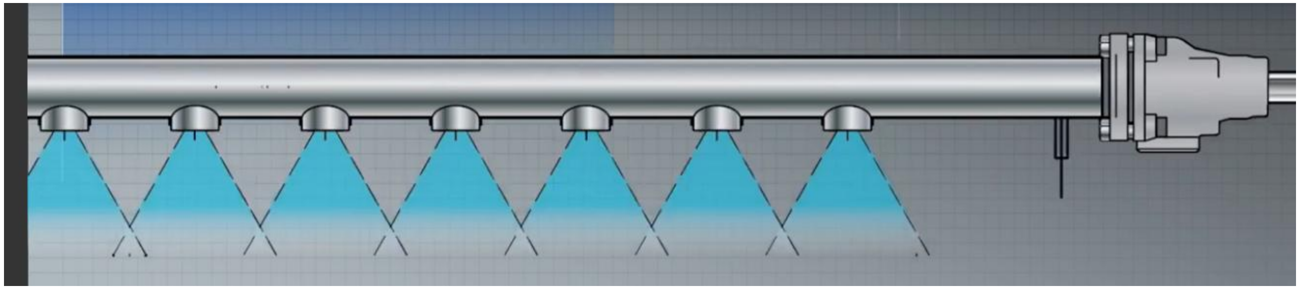
Plant: *NxN* constant model representing an idealized manufacturing process with cross-coupling.

Issue: The plant is non-diagonal and inversion-based control is sensitive to actuator uncertainty.

Controllers: Compare an inversion-based controller and a DK-synthesis controller that accounts for uncertainty.

File: mimoSolve.m

Ref: Image from www.spray.com



Idealized (Constant) Plant Model

Represent the idealized manufacturing process by constant N-by-N real matrix. The plant is *not* diagonal.

```
N=5;
G = toeplitz([1;.75;.40;zeros(N-3,1)],[1 .75 .40 zeros(1,N-3)]);
G = ss(G)
```

1	0.75	0.4	0	0
0.75	1	0.75	0.4	0
0.4	0.75	1	0.75	0.4
0	0.4	0.75	1	0.75
0	0	0.4	0.75	1

This constant model neglects any actuator dynamics.

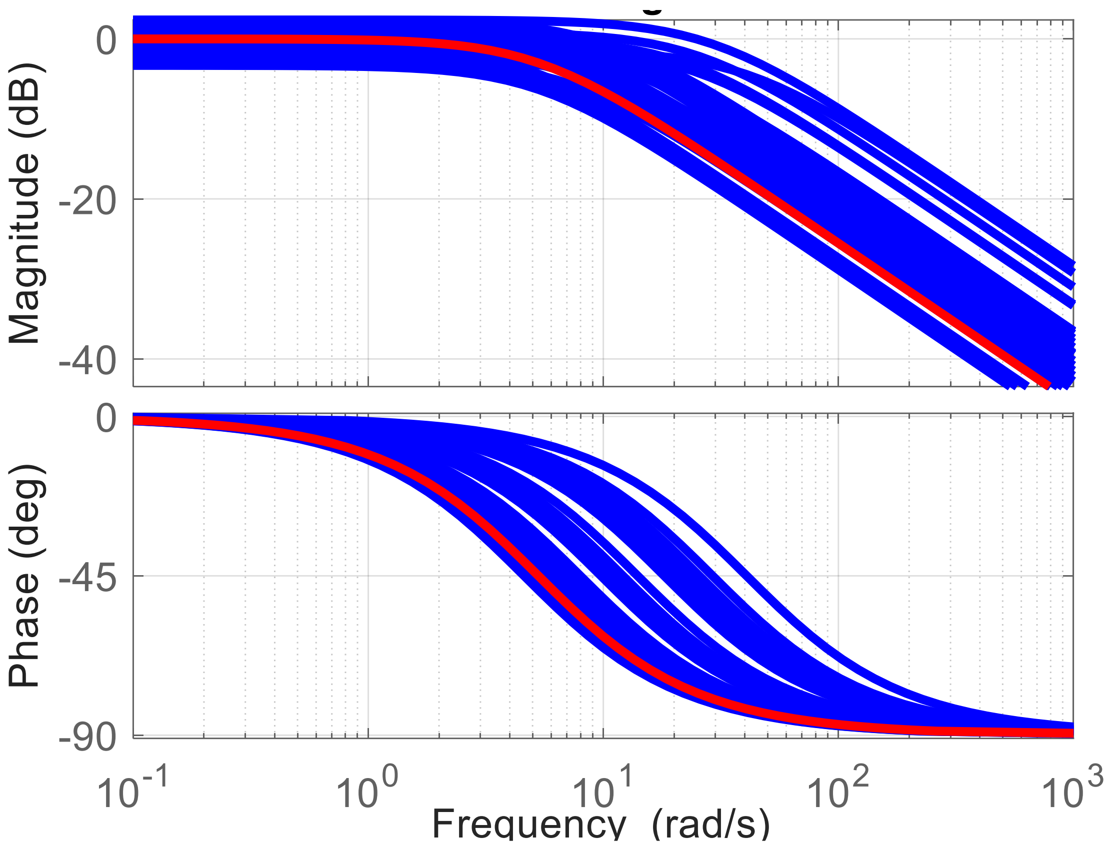
Actuator Models

Assume each channel is modeled by a first-order actuator:

$$A(s) = diag(A_1(s), A_2(s), \dots, A_N(s))$$

where for $i = 1, \dots, N$

- $A_i(s) = \frac{\gamma_i}{\tau_i s+1}$
- $\tau_i \in [0.022, 0.22]$
- $\gamma_i \in [0.67, 1.33]$



Inversion-Based Controller

Use a simple inversion-based controller (neglecting actuator dynamics) with a 1rad/sec along each channel:

$$K_{inv}(s) = \frac{1}{s}G^{-1}$$

The loop without actuator dynamics is:

$$L_{inv}(s) = GK_{inv}(s) = \frac{1}{s}I_N$$

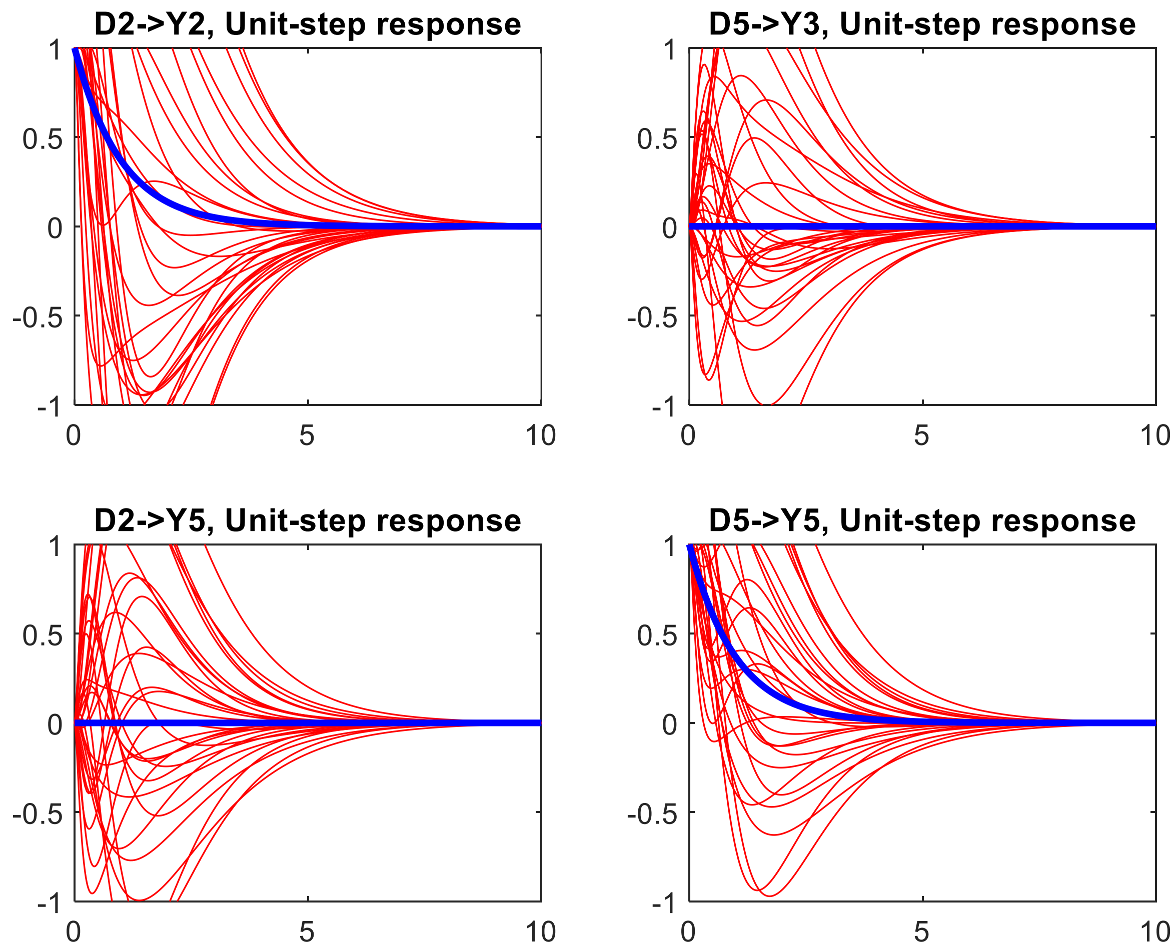
However, the actuator dynamics will prevent perfect inversion and the actual loop will be:

$$L_{inv}(s) = GA(s)K_{inv}(s) = \frac{1}{s}GA(s)G^{-1}$$

Closed-Loop Step Disturbance Responses

Plots show a few typical responses for the nominal (blue) and with samples of uncertain actuators.

These responses show the sensitivity of the inversion controller.



Uncertainty Weight For Actuators

Actuator dynamics:

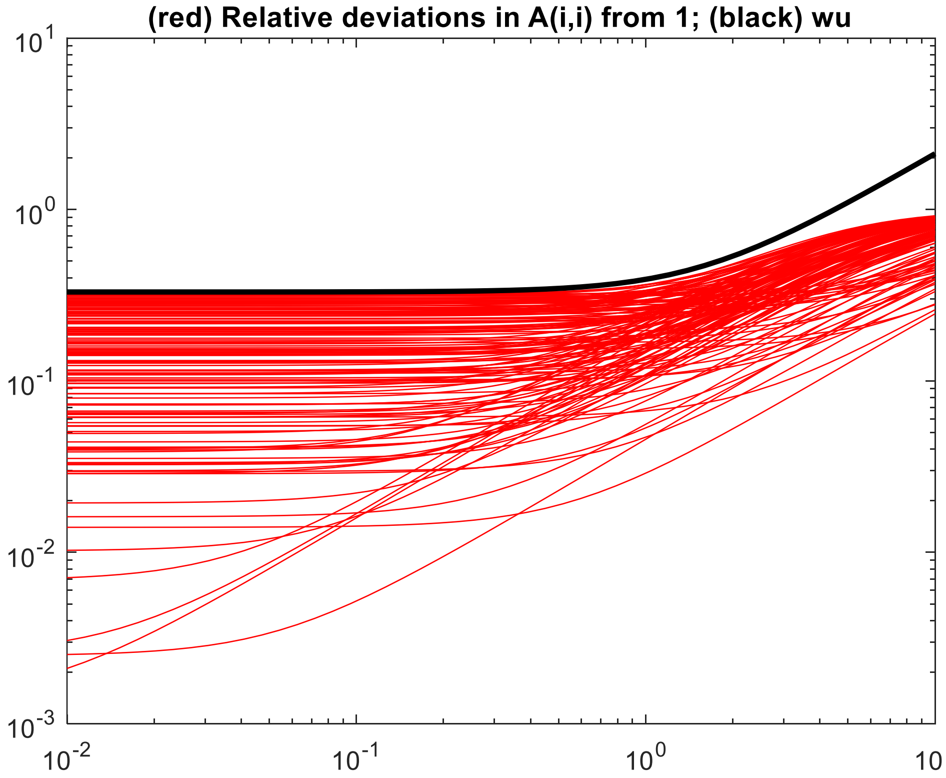
- Nominal actuator is A(s) = 1 (neglected actuator dynamics)
- $\tilde{A}(s) = \frac{\gamma}{\tau s+1}$ with $\tau \in [0.022, 0.22]$ and $\gamma \in [0.67, 1.33]$.

We can bound the relative (percent) error in the actuator:

$$\left| \frac{\tilde{A}(j\omega) - A(j\omega)}{A(j\omega)} \right| < |W_U(\omega)|$$

where

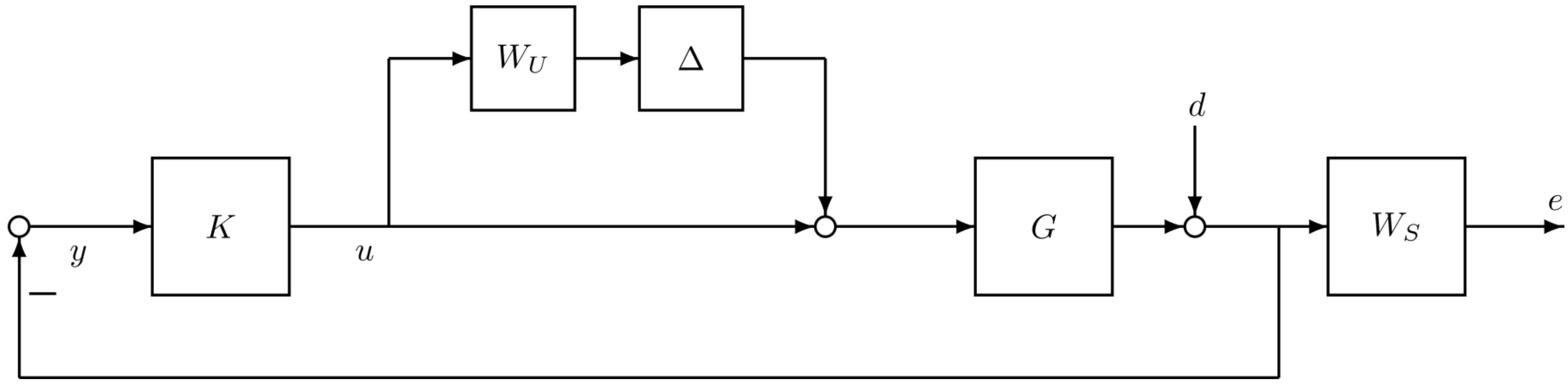
$$W_U(s) = \frac{20\, s + 31.42}{s + 95.22}$$



Robust Control Formulation

Design a controller to stabilize the uncertain plant and minimize the worst-case gain from d to e with

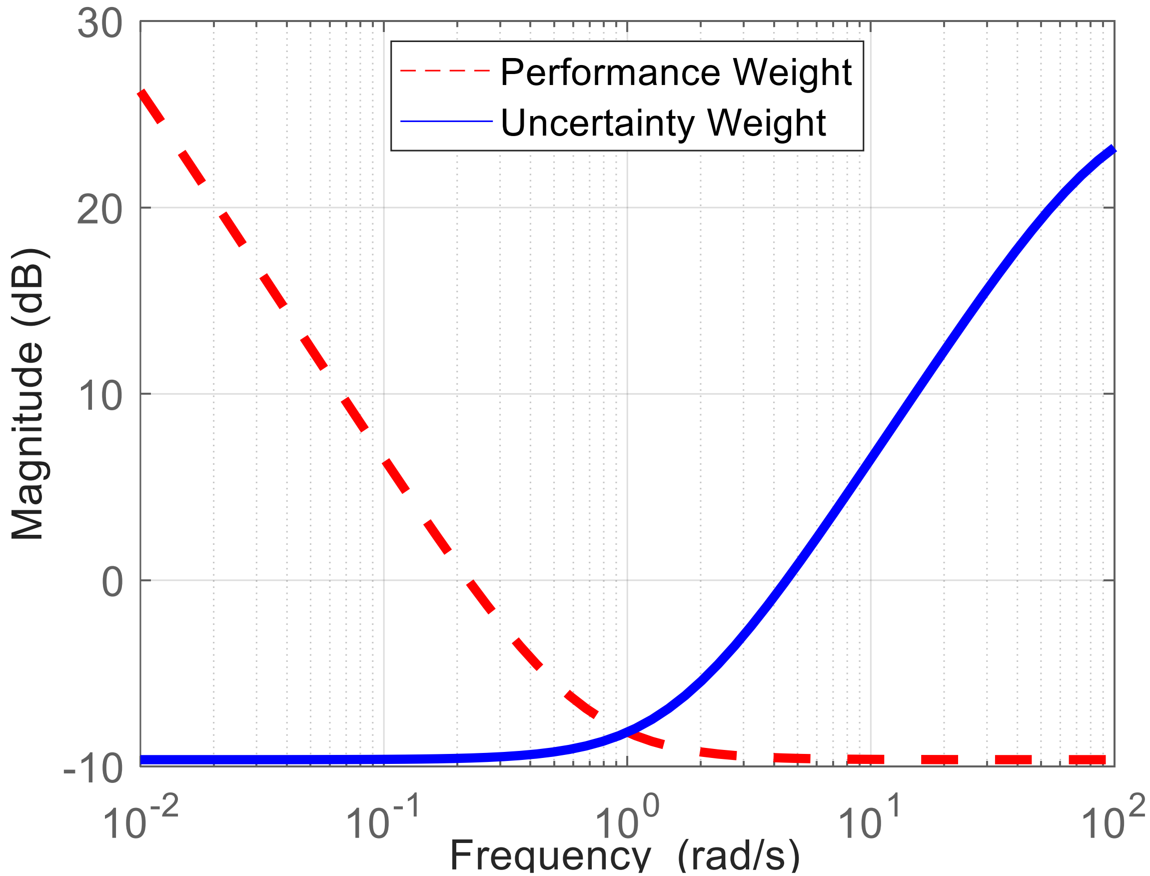
$$W_S(s) = \frac{0.33\,s + 0.2098}{s+0.0021}$$



Robust Control Formulation

Design a controller to stabilize the uncertain plant and minimize the worst-case gain from d to e with

$$W_S(s) = \frac{0.33\, s + 0.2098}{s+0.0021}$$



DK-Iteration Design

```
opt = musynOptions;
opt.FitOrder = [8 2];  % limit each d-scale to be 8th order or less
opt.FrequencyGrid = logspace(-3,3,120);   opt.Maxiter = 12;
[Kmu,Tde,DKinfo] = musyn(P,N,N,opt);
```

```
-----------------------------------------------------------------
            Robust performance              Fit order
-----------------------------------------------------------------
 Iter       K Step      Peak MU      D Fit          D
   1        38.84       21.99       22.28          30
   2        18.11       10.61       10.68          22
   3        5.352       3.863       3.898          34
....
  11         1.942       1.935       1.961          70
  12         1.912       1.908       1.933          72
Best achieved robust performance: 1.91
```

Controller Reduction

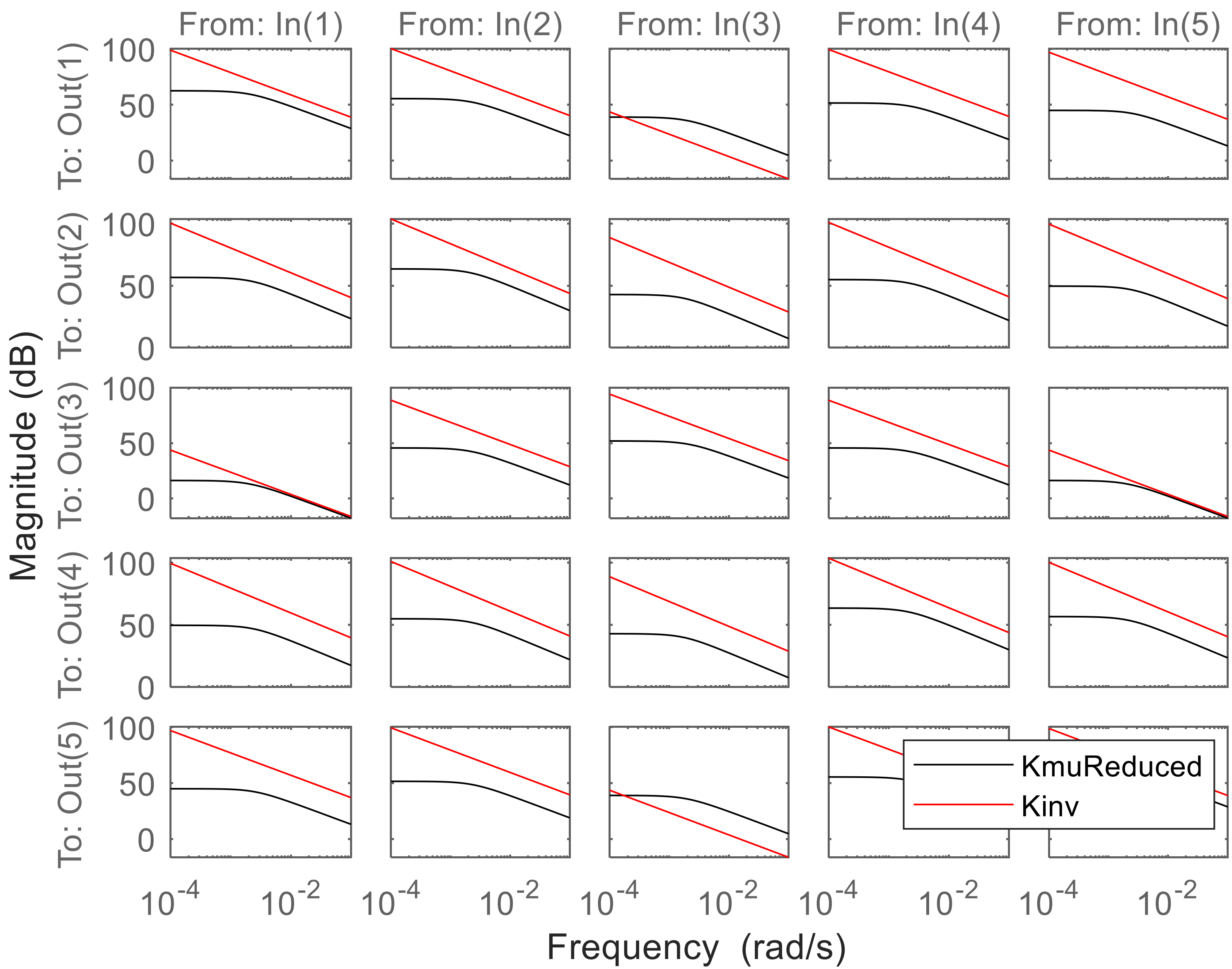
The nominal plant model has 0 states but the iterative process uses dynamic D-scales are used. The final order of $K_{mu}$is 80.

This is quite high but significant model reduction is typically possible with little loss of performance.

Here we reduce to 5 states using balanced truncation and get similar performance

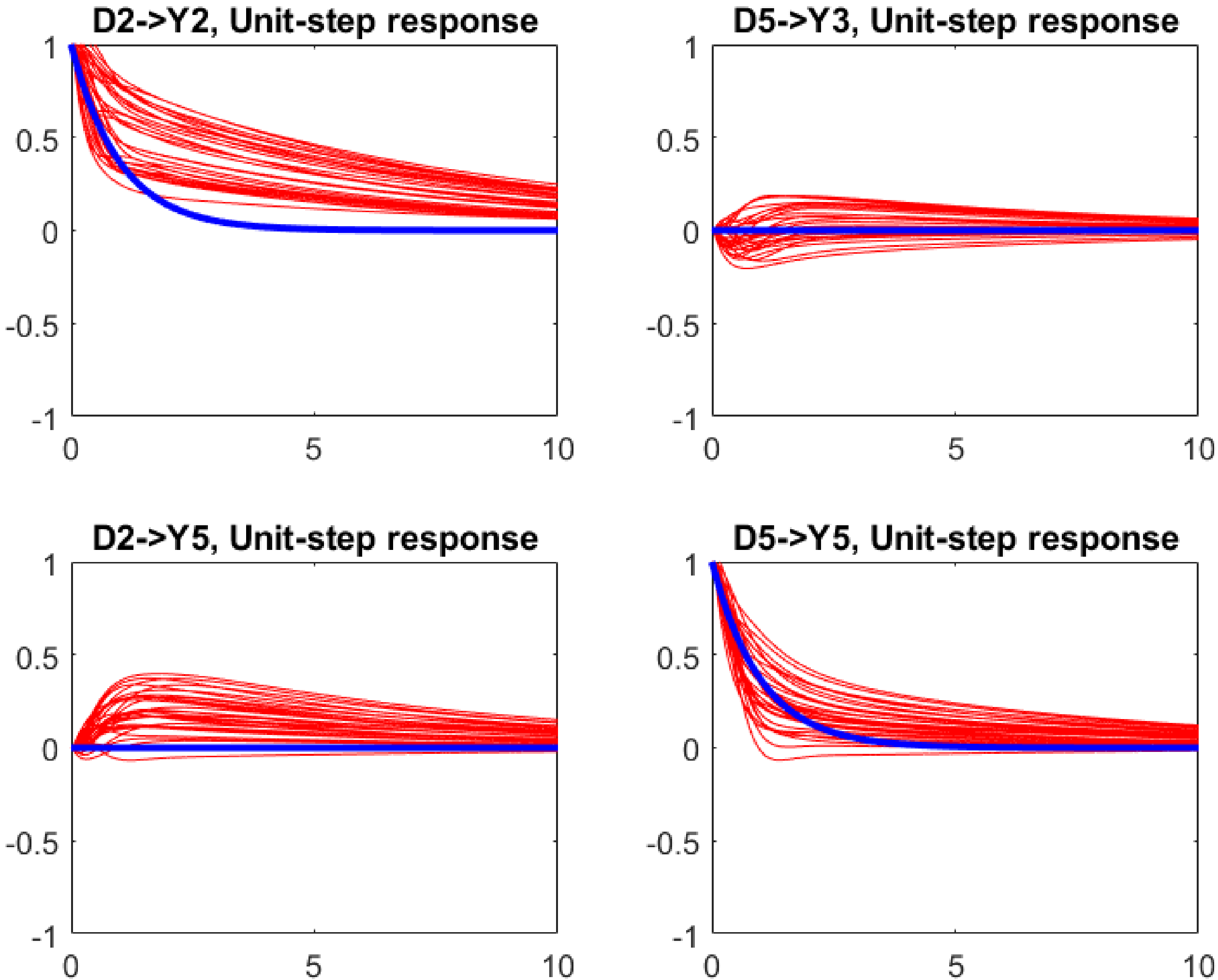
```
stateorder = 5;
Kred = reduce(Kmu,stateorder);
```

Inversion vs. Mu Controller



Closed-Loop Step Disturbance Responses

Plots show a few typical responses for the nominal (blue) and with samples of uncertain actuators.



These responses are more uniform with the mu controller.

Example: Simultaneous Stabilization

Slides: sim_stab_2025_03_21.pdf

Key Takeaways

This lecture focused on designing a controller to optimize the closed-loop performance with an uncertain plant.

This is formulated as a μ −synthesis problem and is solved via DK-iteration alternating between two steps:

1. D-step: Solve for μ upper bound (D-scales) on closed-loop robust performance for a fixed controller.
2. K-step: Solve for optimal controller with fixed D-scales.

Several examples were given to illustrate this method.