Advances in Robust Loopshaping

Richard D. Braatz* Manfred Morari†

Chemical Engineering, 210-41
California Institute of Technology
Pasadena, CA 91125

Sigurd Skogestad

Chemical Engineering
University of Trondheim, NTH
N-7034 Trondheim, Norway

25 SEP. 92

Prepared for presentation at the November 1992 AIChE meeting
"Advances in Process Control" Session

Abstract

Robust performance is said to be achieved if the performance specifications are met for all plants in a specified set. Classical loopshaping was developed decades ago by Bode to design for robust performance for SISO systems, where the uncertainty can be represented as a single complex Δ-block, and the sole performance specification is an upper bound on the closed loop sensitivity. Uncertainty and performance specifications are often not so simple—control problems often involve multiple performance specifications, and uncertainty is sometimes more conveniently described as real parameter variations. Also, it is important for multivariable systems that uncertainty may be present at different locations, for example actuator uncertainty is located at the input of the plant whereas sensor uncertainty is located at the output of the plant.

Previous work by the authors extended classical loopshaping to multiple uncertainty and performance blocks and to the design of decentralized controllers. The authors refer to this more general loopshaping technique as *robust loopshaping*. In this article, we extend *robust loopshaping* to handle mixed real and complex uncertainties, and to more advanced performance specifications. We show how to directly design decentralized controllers such that stability or specified performance for the remaining system remain satisfied when actuators/sensors become faulty or fail. The controllers can also be designed to remain stable with arbitrary detuning of control loops. To the authors' knowledge, this is the first controller synthesis method

*supported by the Fannie and John Hertz Foundation

†To whom all correspondence should be addressed: phone (818)356-4186, fax (818)568-8743, e-mail MM@IMC.CALTECH.EDU

to design controllers with this form of failure-tolerance for systems with general performance and uncertainty structure.

We also discuss the advantages and disadvantages of loopshaping the open loop transfer function or the controller instead of the closed loop transfer functions.

1 Introduction

Loopshaping involves directly specifying a transfer function that parametrizes the controller based on magnitude bounds on the transfer function. These bounds are either necessary conditions or sufficient conditions so that the closed loop system satisfies desired stability and performance specifications. Examples of transfer functions that parametrize the controller include the sensitivity $S = (I + PK)^{-1}$, the complementary sensitivity $H = PK(I + PK)^{-1}$, and the open loop transfer function $L = PK$. The controller K is then calculated from the specified transfer function.

Robust performance is said to be achieved if the performance specifications are met for all plants in a specified set. Controller design methods can be classified as being either optimization methods, or not. The optimization approach involves minimizing an objective function over the set of stabilizing controllers. The optimization objective for robust control is to minimize the robust performance measure μ over the set of all stabilizing controllers, where μ is a function of the nominal model, the controller, the model uncertainty, and the performance specifications. How to solve this optimization problem for centralized controllers is an open question—the ad-hoc "DK-iteration" method proposed by Doyle [8] is the only method of tackling the optimization to date. The DK-iteration method assumes that all uncertainties are complex, involves iterative optimization, has many fragile steps, and produces high order controllers. The DK-iteration method cannot be used effectively to design decentralized controllers, or controllers that are tolerant to failures in actuators or sensors.

Loopshaping can be classified as a non-optimization approach. The advantages of loopshaping over optimization approaches are that: 1) the controller can be kept simple, 2) decentralized controllers can be designed, and 3) the properties of interest to the engineer are often directly in terms of the designed loopshape.

The technique of loopshaping was introduced by Bode [2] for the design of feedback amplifiers. Doyle et al. [9] review classical loopshaping, where the system is SISO, the uncertainty can be represented as a single complex Δ-block, and the sole performance specification is an upper bound on the closed loop sensitivity. Uncertainty and performance specifications are often not so simple—control problems often involve multiple performance specifications, and uncertainty is sometimes more conveniently described as real parameter variations. Also, it is important for multivariable systems that uncertainty may be present at different locations, for example actuator uncertainty is located at the input of the plant whereas sensor uncertainty is located at the output of the plant.

Skogestad and Morari [18, 19] extended classical loopshaping to multiple uncertainty and performance blocks and to the design of decentralized controllers, where they used the parametrizing transfer functions to be the sensitivity and complementary sensitivity. The method had the drawback of only giving sufficient conditions for robust performance. Braatz et al. [5] extended the method by deriving necessary bounds for meeting robust performance. Especially interesting is when the necessary and the sufficient bounds are very close to each other—in this case the bounds are essentially necessary *and* sufficient.

This paper advances the work of Braatz et al. [5]. We extend the work to handle mixed real and complex uncertainties, and more advanced performance specifications than can be addressed directly through "DK-iteration". We show how to directly design decentralized controllers such that stability or specified performance for the remaining system remain satisfied when actuators/sensors become faulty or fail. The controllers can also be designed to remain stable with arbitrary detuning of control loops. To the authors' knowledge, this is the first controller synthesis method to design controllers with this form of failure-tolerance for systems with general performance and uncertainty structure.

We develop an additional bound which is used when loopshaping the open loop transfer function L or the controller K, namely a sufficient lower bound on a transfer function for robust performance to be achieved. Skogestad and Morari [18, 19] looked at jointly loopshaping the sensitivity and complementary sensitivity—we discuss in detail the advantages and disadvantages of loopshaping the open loop transfer function or the controller instead.

Organization The remainder of this report is organized as follows. First we review the structured singular value framework and some additional background material. Second, we present the robust loopshaping bounds for mixed real/complex uncertainties. We then compare and contrast loopshaping via open loop transfer functions with loopshaping via closed loop transfer functions. As an example we design a PD controller for a DC motor with negligible viscous damping by loopshaping the open loop transfer function L, and compare this with a controller designed via loopshaping S and H. Next we discuss how to design robust controllers via loopshaping to meet gain and phase margin specifications, and give an example of this method. Then we show how to design for multiple performance specifications using loopshaping, and how to design for fault and failure tolerance. We give necessary and sufficient tests for versions of fault and failure tolerance that have been defined previously in the literature. We then design a failure and fault tolerant decentralized controller for a high-purity distillation column. The paper finishes with conclusions and some ideas for future work.

2 Background

2.1 Robust Performance

The goal of any controller design is that the overall system is stable and satisfies some minimum performance requirements. These requirements should be satisfied at least when the controller is applied to the *nominal* plant, that is, we require nominal stability and nominal performance.

In practice the real plant $\hat{P}$ is not equal to the model P. The term *robust* is used to indicate that some property holds for a set Π of possible plants $\hat{P}$ as defined by the uncertainty description. In particular, by *robust stability* we mean that the closed loop system is stable for all $\hat{P} \in \Pi$. By *robust performance* we mean that the performance requirements are satisfied for all $\hat{P} \in \Pi$. Performance is commonly defined in robust control theory using the H_∞-norm of some transfer function of interest.

Definition 2.1 *The closed loop system exhibits nominal performance if*

$$||\Sigma||_\infty \equiv \sup_\omega \overline{\sigma}(\Sigma) < 1. \tag{1}$$

Definition 2.2 *The closed loop system exhibits robust performance if*

$$||\hat{\Sigma}||_\infty \equiv \sup_\omega \overline{\sigma}(\hat{\Sigma}) < 1, \qquad \forall \hat{P} \in \Pi. \tag{2}$$

For example, for rejection of disturbances at the plant output, Σ would be the weighted sensitivity

$$\begin{aligned} \Sigma &= W_1 S W_2, \quad S = (I + PK)^{-1}, \\ \hat{\Sigma} &= W_1 \hat{S} W_2, \quad \hat{S} = (I + \hat{P}K)^{-1}. \end{aligned} \tag{3}$$

In this case, the input weight W_2 is often equal to the disturbance model. The output weight W_1 is used to specify the frequency range over which the sensitivity function should be small and to weigh each output according to its importance. K is the transfer function of the controller.

Doyle [7] derived the *structured singular value*, μ, to test for robust performance. To use μ we must model the uncertainty (the set Π of possible plants $\hat{P}$) as norm bounded perturbations (Δ_i) on the nominal system. Through weights each perturbation is normalized to be of size one:

$$||\Delta_i||_\infty \leq 1. \tag{4}$$

The perturbations, which may occur at different locations in the system, are collected in the block-diagonal matrix Δ_U (the U denotes uncertainty)

$$\Delta_U = \operatorname{diag}\{\Delta_i\} \tag{5}$$

and the system is arranged to match the left block diagram in **Figure 1.** The interconnection matrix M in Figure 1 is determined by the nominal model (P), the size and nature of the uncertainty, the performance specifications, and the controller (K).

Without loss of generality we assume that each Δ_i and M is square [14]. The definition of μ is:

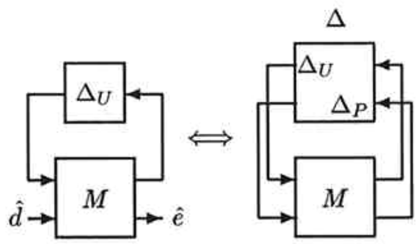

Figure 1: Robust Performance and the $M - \Delta$ block structure

Definition 2.3 *Let $M \in \mathbf{C}^{n\times n}$ be a square complex matrix and define the set $\boldsymbol{\Delta}$ of block-diagonal perturbations by*

$$\boldsymbol{\Delta} \equiv \left\{ \operatorname{diag}\left\{\delta_1^r I_{r_1}, \cdots, \delta_k^r I_{r_k}, \delta_{k+1}^c I_{r_{k+1}}, \cdots, \delta_m^c I_{r_m}, \Delta_{r_{m+1}}, \cdots, \Delta_{r_l}\right\} \middle| \delta_i^r \in \mathbf{R}, \delta_i^c \in \mathbf{C}, \Delta_{r_i} \in \mathbf{C}^{r_i \times r_i}, \sum_{i=1}^{l} r_i = n \right\}. \tag{6}$$

Then $\mu_{\boldsymbol{\Delta}}(M)$ (the structured singular value with respect to the uncertainty structure $\boldsymbol{\Delta}$) is defined as

$$\mu_{\boldsymbol{\Delta}}(M) \equiv \begin{cases} 0 \text{ if there does not exist } \Delta \in \boldsymbol{\Delta} \text{ such that } \det(I - M\Delta) = 0, \\ \left[\min\limits_{\Delta\in\boldsymbol{\Delta}} \left\{\overline{\sigma}(\Delta)\,|\det(I - M\Delta) = 0\right\}\right]^{-1} \quad \text{otherwise.} \end{cases} \tag{7}$$

Partition M in Fig. 1 to be compatible with $\Delta = \operatorname{diag}\{\Delta_U, \Delta_P\}$:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}. \tag{8}$$

The following are tests for robust stability and robust performance [7]

Theorem 2.4 *The closed loop system exhibits robust stability for all $\|\Delta_U\|_\infty \leq 1$ if and only if the closed loop system is nominally stable and*

$$\mu_{\boldsymbol{\Delta}_U}(M_{11}(j\omega)) < 1 \qquad \forall\omega. \tag{9}$$

Theorem 2.5 *The closed loop system exhibits robust performance for all $\|\Delta_U\|_\infty \leq 1$ if and only if the closed loop system is nominally stable and*

$$\mu_{\boldsymbol{\Delta}}(M(j\omega)) < 1 \qquad \forall\omega, \tag{10}$$

where $\boldsymbol{\Delta} = \operatorname{diag}\{\Delta_U, \Delta_P\}$, and Δ_P is a full square matrix with dimension equal to the number of outputs (the subscript P denotes performance).

Multiple performance objectives can be tested similarly using block-diagonal Δ_P. Note that the issue of robust stability is simply a special case of robust performance.

It is a key idea that μ is a *general* analysis tool for determining robust performance. Any system with uncertainty adequately modeled as in (4) can be put into $M - \Delta_U$ form, and robust stability and robust performance can be tested using (9) and (10). Standard programs calculate the M and Δ [1], given the transfer functions describing the system components and the location of the uncertainty and performance blocks Δ_i.

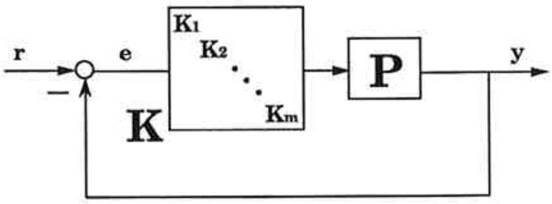

Figure 2: General decentralized control structure

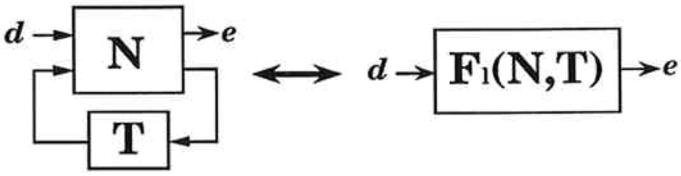

Figure 3: Definition of the linear fractional transformation $F_l(N,T)$.

2.2 Decentralized Control

Decentralized control involves using a diagonal or block-diagonal controller (see Figure 2)

$$K = \text{diag}\{K_i\}. \tag{11}$$

This includes controllers that can be made block-diagonal by reordering the measured variables and manipulated variables.

Some reasons for using a decentralized controller are

- tuning and retuning is simple
- they are easy to understand
- they are easy to make failure tolerant
- implementation and maintenance is simpler

These reasons explain the predominance of decentralized controllers in applications.

The design of a decentralized control system involves two steps. First the *control structure* must be selected. This involves the choosing of the actuators and sensors and the pairings between the chosen actuators and sensors. Methods of selecting the control structure for uncertain systems are presented in [5, 14]. The second step is the ***design*** of each SISO (or block) controller K_i. These controllers can be designed using the robust loopshaping method discussed in this paper.

2.3 Linear Fractional Transformations

We will define the special linear interconnection structure called the linear fractional transformation (LFT). The lower LFT denoted $F_l(N,T)$ is defined by (see Figure 3)

$$F_l(N,T) = N_{11} + N_{12}T(I - N_{22}T)^{-1}N_{21}. \tag{12}$$

$F_l(N,T)$ is well-defined if and only if the inverse of $I - N_{22}T$ exists. A superscript is sometimes used on N, *e.g.* N^T, to denote that N depends on the choice of T.

The subscript l on F_l is used to denote that the lower loop of N is closed by T. When the upper loop is closed, the transfer function between inputs and outputs is the LFT $F_u(N,T) = N_{22}+N_{21}T(I-N_{11}T)^{-1}N_{12}$.

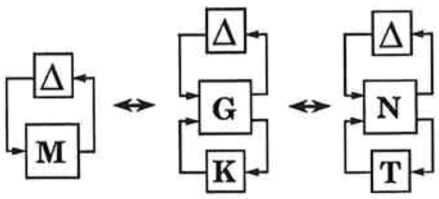

Figure 4: Equivalent representations of system M with perturbation Δ. T is chosen to be a parametrization of the controller K.

2.4 Sensitivity and Complementary Sensitivity

The *complementary sensitivity* H and the *sensitivity* S are defined by

$$H = PK(I + PK)^{-1}, \qquad\qquad S = (I + PK)^{-1}. \tag{13}$$

These functions are useful in the loopshaping design of SISO controllers. For loopshaping design, H and S will be considered as parametrizations of the controller K. Note that $S + H = I$.

Define $\tilde{P}$ to be the block-diagonal part of the plant corresponding to the block structure of the decentralized controller K. If

$$K = \mathrm{diag}\{K_1, K_2, \cdots, K_m\}, \tag{14}$$

then

$$\tilde{P} = \mathrm{diag}\{P_{11}, P_{22}, \cdots, P_{mm}\}. \tag{15}$$

Define the *block-diagonal complementary sensitivity* by $\tilde{H} = \tilde{P}K(I+\tilde{P}K)^{-1}$ and the *block-diagonal sensitivity* by $\tilde{S}(I + \tilde{P}K)^{-1}$. The block-diagonal complementary sensitivity and block-diagonal sensitivity are useful in the loopshaping design of decentralized controllers. For decentralized controller design, $\tilde{H}$ and $\tilde{S}$ will be the parametrizations of the block-diagonal controller K. Note that $\tilde{H} + \tilde{S} = I$.

2.5 Parametrize Controller as a Function of T

To use the theorems derived in this paper an LFT of M must be found in terms of T (see Fig. 4), where T parameterizes the controller we are to design. In many cases, this is easily done by inspection. In other cases, a procedure in [18] is used. For loopshaping, T is usually chosen to be H, S, L, K, $\tilde{H}$, $\tilde{S}$, or $\tilde{L}$ (see Braatz et al. [5] for derivation of the equations that follow).

To get N^H, begin with the interconnection structure in terms of G and K. The generalized plant G is determined by the nominal model, the location and magnitude of the uncertainties, and the performance specifications. The generalized plant G is found directly by rearranging the system's block diagram.[1] We can calculate N for $T = H$ (denoted as N^H) directly from G:

$$N^H = \begin{bmatrix} G_{11} & G_{12}P^{-1} \\ G_{21} & 0 \end{bmatrix}. \tag{16}$$

For $T = S$, L, and K, respectively, we have

$$N^S = \begin{bmatrix} G_{11} + G_{12}P^{-1}G_{21} & -G_{12}P^{-1} \\ G_{21} & 0 \end{bmatrix}, \tag{17}$$

$$N^L = \begin{bmatrix} G_{11} & G_{12}P^{-1} \\ G_{21} & G_{22}P^{-1} \end{bmatrix}, \tag{18}$$

$$N^K = G, \tag{19}$$

[1] The subroutine *sysic* does this in μ-tools [1].

and for decentralized control,

$$N^{\tilde{H}} = \begin{bmatrix} G_{11} & G_{12}\tilde{P}^{-1} \\ G_{21} & I - P\tilde{P}^{-1} \end{bmatrix}, \tag{20}$$

$$N^{\tilde{S}} = \begin{bmatrix} G_{11} + G_{12}P^{-1}G_{21} & -G_{12}P^{-1} \\ \tilde{P}P^{-1}G_{21} & I - \tilde{P}P^{-1} \end{bmatrix}, \tag{21}$$

$$N^{\tilde{L}} = \begin{bmatrix} G_{11} & G_{12}\tilde{P}^{-1} \\ G_{21} & G_{22}\tilde{P}^{-1} \end{bmatrix}. \tag{22}$$

A simple program can be written that calculates N^H, N^S, N^L, N^K, $N^{\tilde{H}}$, $N^{\tilde{S}}$, and $N^{\tilde{L}}$ given the transfer functions describing the system components and the location of the uncertainty blocks Δ_i.

3 Robust Loopshaping Bounds

Controllers which satisfy robust performance can be designed via robust loopshaping. To perform robust loopshaping, the robust performance conditions are expressed as norm bounds on the transfer function T.

A sufficient condition was derived by Skogestad and Morari [18] for robust performance based on $\overline{\sigma}(T)$ being small enough. It was shown that this bound on $\overline{\sigma}(T)$ was the tightest bound possible, i.e. if we have a T_1 with $\overline{\sigma}(T_1)$ larger than the bound given in [18], then there exists a T_2 with $\overline{\sigma}(T_2) = \overline{\sigma}(T_1)$ where T_2 does not meet robust performance. The tightest *necessary* upper and lower bounds were derived in [5]. $\overline{\sigma}(T)$ must be between the upper and lower necessary bounds. Comparing the necessary and sufficient bounds gives a measure of the conservativeness of loopshaping design. When the necessary bound and the sufficient bound are very close to each other, we have essentially necessary *and* sufficient bounds for robust performance in terms of $\overline{\sigma}(T)$. Below we also present an additional bound, the sufficient lower bound, which is useful when loopshaping design using open loop transfer functions (e.g. K and L).

Consider a system in $M-\Delta$ form as shown in Figure 4. The interconnection structures in Figure 4 are equivalent. M is written as a linear fractional transformation of the transfer function of interest, namely T. We define the set of all norm-bounded perturbations

$$\gamma\mathbf{B\Delta}_T \equiv \{\Delta_T \,|\Delta_T \in \mathbf{\Delta}_T, \overline{\sigma}(\Delta_T) \leq \gamma\}, \tag{23}$$

and also it near-complement

$$\gamma\overline{\mathbf{B\Delta}}_T \equiv \{\Delta_T \,|\Delta_T \in \mathbf{\Delta}_T, \overline{\sigma}(\Delta_T) \geq \gamma\}. \tag{24}$$

The robust loopshaping bounds are briefly described below (see Braatz et al. [5] for proofs, a thorough description, and several examples of the use of the bounds for designing SISO and decentralized controllers).

Theorem 3.1 (Sufficient Upper Bound for Robust Performance [18, 19]) *Let $M = F_l(N,T) = N_{11} + N_{12}T(I-N_{22}T)^{-1}N_{21}$, let k be a given constant, and define*

$$f(c_T) = \max_{\Delta_T \in c_T \mathbf{B\Delta}_T} \mu_{\mathbf{\Delta}}(F_l(N,\Delta_T)). \tag{25}$$

Assume

$$\begin{aligned} (i) &\quad \det(I - N_{22}T) \neq 0, \\ (ii) &\quad f(0) = \mu_{\mathbf{\Delta}}(N_{11}) < k, \text{ and} \\ (iii) &\quad f(\infty) > k. \end{aligned} \tag{26}$$

Let c_T^{su} solve

$$f(c_T^{su}) = k. \tag{27}$$

Then $\mu_{\mathbf{\Delta}}(M) < k$ if

$$\overline{\sigma}(T) < c_T^{su}. \tag{28}$$

Remark 3.2 (Remarks Regarding Assumptions) *Assumption* (i) *will hold for any well-posed problem. If assumption* (iii) *does not hold, then* any T *will give* $\mu_{\boldsymbol{\Delta}}(M) < k$*—the uncertainty and performance weights would have to be very weak for this to be the case. Assumption* (ii) *may or may not hold. For reasonable choices of uncertainty and performance weights, assumption* (ii) *will hold for low frequencies when* $T = S$ *or* $\tilde{S}$ *and will hold for high frequencies when* $T = H$, $\tilde{H}$, L, *or* K. *This will be illustrated in more detail later.*

Remark 3.3 (Remarks Regarding Calculation) *Theorem 3.1 is a restatement of a result in [18, 19]. In fact, Skogestad and Morari [18] show that Thm. 3.1 remains valid if* $f(c_T)$ *is replaced by*

$$\hat{f}(c_T) \equiv \mu_{\begin{bmatrix} \boldsymbol{\Delta} & \\ & \boldsymbol{\Delta}_{\mathrm{T}} \end{bmatrix}} \left(\begin{bmatrix} N_{11} & N_{12} \\ kc_T N_{21} & kc_T N_{22} \end{bmatrix} \right). \tag{29}$$

This means that the sufficient upper bound can be calculated through iterative μ *calculations. Actually, the sufficient upper bound can be calculated via one* scaled μ *calculation, as is discussed by Braatz et al [5].*

Theorem 3.4 (Sufficient Lower Bound for Robust Performance) *Let* $M = F_l(N,T) = N_{11}+N_{12}T(I-N_{22}T)^{-1}N_{21}$, *let* k *be a given constant, and define*

$$e(c_T) = \max_{\Delta_T \in c_T \overline{\mathbf{B}\boldsymbol{\Delta}}_T} \mu_{\boldsymbol{\Delta}}(F_l(N,\Delta_T)). \tag{30}$$

Assume

$$\begin{array}{rl} (i) & \det(I - N_{22}T) \neq 0, \\ (ii''') & e(0) = \max_{\Delta_T \in \boldsymbol{\Delta}_T} \mu_{\boldsymbol{\Delta}}(F_l(N,\Delta_T)) > k, \text{ and} \\ (iii''') & e(\infty) < k. \end{array} \tag{31}$$

Let c_T^{sl} *solve*

$$e(c_T^{sl}) = k. \tag{32}$$

Then $\mu_{\boldsymbol{\Delta}}(M) < k$ *if*

$$\overline{\sigma}(T) > c_T^{sl}. \tag{33}$$

Remark 3.5 (Remarks Regarding Assumptions) *Assumption* (i) *will hold for any well-posed problem. If assumption* (ii''') *does not hold, than* any T *will give* $\mu_{\boldsymbol{\Delta}}(M) < k$*—the performance specifications are trivially achieved in this case. For reasonable choices of uncertainty and performance weights, assumption* (iii''') *will hold for low frequencies when loopshaping* K *or* L. *This bound never exists when loopshaping with closed loop transfer functions (with reasonably chosen weights).*

Remark 3.6 (Remarks Regarding Calculation) *The sufficient lower bound can be calculated through a scaled* μ *calculation whenever* $\boldsymbol{\Delta}_T$ *is repeated scalar. This will happen when designing an SISO controller, or a decentralized controller for robust performance.*

To show this, we need the following lemma:

Lemma 3.7 (LFT of Inverse) *Consider an LFT in terms of* A, $F_l(N,A)$. *Define*

$$\hat{N} = \begin{bmatrix} N_{11} - N_{12}N_{22}^{-1}N_{21} & N_{12}N_{22}^{-1} \\ -N_{22}^{-1}N_{21} & N_{22}^{-1} \end{bmatrix}. \tag{34}$$

Then the following equality holds:

$$F_l(N,A) = F_l(\hat{N}, A^{-1}). \tag{35}$$

Proof: *The equality follows from the definition of the lower LFT and some simple algebraic manipulation. QED.*

If $\mathbf{\Delta}_T$ is repeated scalar, then

$$
\begin{aligned}
e(c_T) &\equiv \max_{\Delta_T \in c_T \overline{\mathbf{B\Delta}_T}} \mu_{\mathbf{\Delta}}(F_l(N, \Delta_T)) && (36)\\
&= \max_{\delta I \in c_T \overline{\mathbf{B\Delta}_T}} \mu_{\mathbf{\Delta}}(F_l(N, \delta I)) && (37)\\
&= \max_{(1/\delta) I \in c_T \mathbf{B\Delta}_T} \mu_{\mathbf{\Delta}}\left(F_l\left(\hat{N}, \delta^{-1} I\right)\right) && (38)\\
&= \max_{\hat{\Delta}_T \in c_T \mathbf{B\Delta}_T} \mu_{\mathbf{\Delta}}\left(F_l\left(\hat{N}, \hat{\Delta}_T\right)\right) && (39)\\
&\equiv f(c_T). && (40)
\end{aligned}
$$

Thus the method for calculating the sufficient upper bound on $\overline{\sigma}(T)$ discussed in Remark 3.3 can also be used to calculate the sufficient lower bound on $\overline{\sigma}(T)$ whenever $\mathbf{\Delta}_T$ is repeated scalar. This fact was used by Hovd [11] to design decentralized controllers via loopshaping the IMC filter parameter.

Theorem 3.8 (Necessary Upper Bound for Robust Performance) *Let $M = F_l(N,T) = N_{11} + N_{12}T(I - N_{22}T)^{-1}N_{21}$, let k be a given constant, and define*

$$g(c_T) = \min_{\Delta_T \in c_T \overline{\mathbf{B\Delta}_T}} \mu_{\mathbf{\Delta}}(F_l(N, \Delta_T)). \tag{41}$$

Assume

$$
\begin{aligned}
(i) &\quad \det(I - N_{22}T) \neq 0,\\
(ii') &\quad g(0) = \min_{\Delta_T \in \mathbf{\Delta}_T} \mu_{\mathbf{\Delta}}(F_l(N, \Delta_T)) < k, \text{ and}\\
(iii') &\quad g(\infty) > k.
\end{aligned}
\tag{42}
$$

Let c_T^{nu} solve

$$g(c_T^{nu}) = k. \tag{43}$$

Then $\mu_{\mathbf{\Delta}}(M) < k$ only if

$$\overline{\sigma}(T) < c_T^{nu}. \tag{44}$$

Remark 3.9 (Remarks Regarding Assumptions) *Assumption (i) will hold for any well-posed problem. If assumption (ii′) does not hold, than* no *T will give $\mu_{\mathbf{\Delta}}(M) < k$—when T parametrizes the controller K, this implies that no controller with the given structure exists that will achieve robust performance. If assumption (iii′) does not hold, then the optimization (41) is too conservative to give a useful necessary upper bound on T. For reasonable choices of uncertainty and performance weights, assumption (iii′) holds for high frequencies when loopshaping K or L. This bound always exists when loopshaping with closed loop transfer functions (with reasonably chosen weights).*

Remark 3.10 (Remarks Regarding Calculation) *Braatz et al [5] show how to calculate the necessary upper bound (and the necessary lower bound discussed below) when Δ_T is repeated scalar. No method currently exists for calculating the necessary bounds for general Δ_T.*

Theorem 3.11 (Necessary Lower Bound for Robust Performance) *Let $M = F_l(N,T) = N_{11} + N_{12}T(I - N_{22}T)^{-1}N_{21}$, let k be a given constant, and define*

$$h(c_T) = \min_{\Delta_T \in c_T \mathbf{B\Delta}_T} \mu_{\mathbf{\Delta}}(F_l(N, \Delta_T)). \tag{45}$$

Assume

$$
\begin{aligned}
(i) &\quad \det(I - N_{22}T) \neq 0,\\
(ii'') &\quad h(\infty) = \min_{\Delta_T \in \mathbf{\Delta}_T} \mu_{\mathbf{\Delta}}(F_l(N, \Delta_T)) < k, \text{ and}\\
(iii'') &\quad h(0) = \mu_{\mathbf{\Delta}}(N_{11}) > k.
\end{aligned}
\tag{46}
$$

Let c_T^{nl} solve

$$h(c_T^{nl}) = k. \tag{47}$$

Then $\mu_{\boldsymbol{\Delta}}(M) < k$ only if

$$\overline{\sigma}(T) > c_T^{nl}. \tag{48}$$

Remark 3.12 (Remarks Regarding Assumptions) *Assumption (i) will hold for any well-posed problem. If assumption (ii″) does not hold, than* no *T will give $\mu_{\boldsymbol{\Delta}}(M) < k$—when T parametrizes the controller K, this implies that no controller with the given structure exists that will achieve robust performance. This will be discussed more later. Assumption (iii″) may or may not hold. For reasonable choices of uncertainty and performance weights, assumption (iii) will hold for high frequencies when $T = S$ or $\tilde{S}$ and will hold for low frequencies when $T = H$, $\tilde{H}$, L, or K. This will be illustrated in more detail later.*

General Remarks

Remark 3.13 *Note that the sufficient upper bound and the necessary lower bound cannot both exist at the same frequency. Actually, when a robustly performing controller exists (assumption (ii") must hold in this case), and provided that robust performance is not trivial to satisfy (so assumption (iii) holds),* exactly *one bound exists for each frequency. Similarly, the sufficient lower bound and the necessary upper bounds cannot both exist at the same frequency. Actually, when a robustly performing controller exists (assumption (ii') must hold in this case), and provided that robust performance is not trivial to satisfy (so assumption (iii"') holds),* exactly *one bound exists for each frequency.*

Remark 3.14 *Many parameterizations T exist for the controller K, for example K can be parameterized by the sensitivity S, the complementary sensitivity H, the block diagonal sensitivity $\tilde{S}$, the block diagonal complementary sensitivity $\tilde{H}$, the open loop transfer function $L = PK$, or just the controller K. Controllers can also be designed via loopshaping the IMC filter F [9] or the IMC filter time constant λ [11].*

Remark 3.15 *The norm bounds on different T's can be combined over different frequency ranges. For example, for $T_1 = S$ and $T_2 = H$, robust performance is achieved if* either *of the conditions $\overline{\sigma}(S(jw)) < c_S$* or *$\overline{\sigma}(H(jw)) < c_H$ is met for each ω. This is important when guaranteeing robust performance when loopshaping closed loop transfer functions.*

Remark 3.16 *The bounds given by each theorem are the tightest bounds possible. For example, if we have a T_1 with $\overline{\sigma}(T_1)$ larger than c_T^{su} defined by Theorem 3.1, then there exists a T_2 with $\overline{\sigma}(T_2) = \overline{\sigma}(T_1)$ where T_2 does not meet robust performance.*

Remark 3.17 *The least conservative bounds are obtained when Δ_T is a repeated scalar block. For this reason, the repeated scalar block (i.e. assuming all loops are identical) is used when designing fully-decentralized controllers for robust performance via loopshaping. When designing controllers to have failure tolerance properties, it can be useful to allow Δ_T to consist of independent 1×1 blocks when calculating sufficient bounds for robust stability. This is explained in Section 7.2.*

Remark 3.18 *It is straightforward to use Lemma 3.7 to derive alternative bounds in terms of $\underline{\sigma}(T)$. In this paper we focus on achieving bounds on $\overline{\sigma}(T)$.*

4 Controller Design via Loopshaping

In robust loopshaping design, the nominal closed loop transfer functions are specified directly based on necessary bounds and sufficient bounds for robust performance. In a previous paper [5], we focused on loopshaping with the sensitivity and the complementary sensitivity for SISO systems, and with the diagonal sensitivity and diagonal complementary sensitivity when designing fully-decentralized controllers.

When designing controllers via loopshaping we need to satisfy separate conditions to guarantee nominal stability. For example, when designing an SISO controller via loopshaping closed loop transfer functions, nominal stability is guaranteed by specifying stable S and H and by satisfying the interpolation conditions [9]

$$H(z_i) = 0 \text{ and } S(z_i) = 1 \text{ for all closed right half plane zeros } z_i, \tag{49}$$

and

$$S(p_i) = 0 \text{ and } H(p_i) = 1 \text{ for all closed right half plane poles } p_i. \tag{50}$$

The interpolation conditions are equivalent to the condition that the right half plane poles and zeros of the plant cannot be cancelled by the controller. These conditions are easy to satisfy when there are few right half plane poles and zeros; when there are more then the Internal Model Control (IMC) method can be used to stabilize the system, and the filter can be designed via loopshaping (for details see [15, 9]). Guaranteeing nominal stability is more difficult in the multivariable case. See [22] for a detailed discussion of stability for multivariable systems. A conservative loopshaping method for guaranteeing nominal stability is in [19].

There are many advantages to designing controllers via loopshaping closed loop transfer functions. One advantage is that the properties of interest to the engineer are specified directly by the nominal closed loop transfer functions. For example, the sensitivity is directly related to the capability of the closed loop system to reject disturbances at the output of the plant. The complementary sensitivity is directly related to the closed loop speed of response and the insensitivity of the output to measurement noise. Thus directly specifying the closed loop transfer functions allows "intuition" in the design procedure. Also, it is easy to specify gain and phase margins when loopshaping the sensitivity S, as will be shown in Example 2b. When designing controllers via loopshaping closed loop transfer functions, robust performance can be guaranteed using sufficient bounds on the sensitivity and complementary sensitivity (e.g. see [5]). As will be shown in Example 2a, robust performance cannot be guaranteed using sufficient bounds on open loop transfer functions.

A simple form is usually chosen for S and H, and the controller is calculated via $K = P^{-1}HS^{-1}$ or $K = \tilde{P}^{-1}\tilde{H}\tilde{S}^{-1}$. A disadvantage of designing controllers via loopshaping *closed loop* transfer functions is that in practice the order of the controller is larger than the order of the plant. It is difficult to design a controller with a specified structure when specifying closed loop transfer functions. The advantage of loopshaping design using *open loop* transfer functions (e.g. L or K) is that the controller complexity (e.g. PID, or low order) is directly specified. It is difficult to do this using other robust controller design methods. For example, the DK-iteration method proposed by Doyle [8] gives controllers of very high order, though the order can be somewhat reduced using model reduction [1].

The next section will show how to use the necessary and the sufficient bounds to design a controller via loopshaping L. Then we show how to design robust controllers with specified gain and phase margins. We then show how to meet multiple performance specifications, including such goals as failure tolerance and fault tolerance.

5 Loopshaping with Open Loop Transfer Functions

In Example 1a we derive loopshaping bounds on L for an SISO plant. The corresponding bounds for K are immediately given by the bounds for L, since $|L| = |P| \cdot |K|$. In Example 2a, we apply these bounds to design a controller for a DC motor with negligible viscous damping.

5.1 Example 1a—Loopshaping L for SISO Plant

Assume that we are interested in disturbance attenuation, then our performance condition is to keep the norm of the sensitivity function $\overline{\sigma}(S) = |S|$ small. If we let our frequency dependent performance bound be $1/|w_P|$, then robust performance is satisfied if $\overline{\sigma}(S) < 1/|w_P|$ for all plants in our uncertainty description. Let the set of possible plants be given in terms of multiplicative uncertainty of magnitude $|w_O|$ (see Fig. 5). Robust performance is satisfied if and only if $\mu_{\mathbf{\Delta}}(M) < 1$ for all frequencies where

$$M = \begin{bmatrix} w_O H & w_O H \\ w_P S & w_P S \end{bmatrix}, \qquad \Delta = \begin{bmatrix} \Delta_O & \\ & \Delta_P \end{bmatrix}, \tag{51}$$

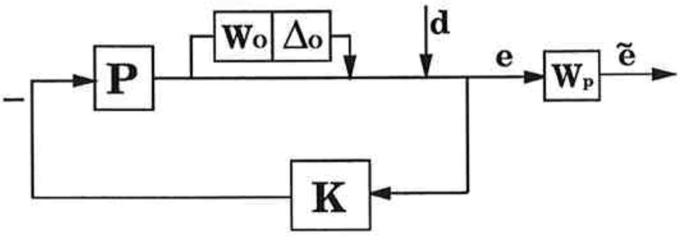

Figure 5: The plant with output uncertainty Δ_O of magnitude $w_O(s)$. Robust performance is satisfied if $\overline{\sigma}\left(w_P(I+\hat{P}K)^{-1}\right) \leq 1$ for all Δ_O with $||\Delta_O||_\infty \leq 1$.

and H is the complementary sensitivity function. The generalized plant G is found from inspection to be

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \left[\begin{array}{cc|c} 0 & 0 & w_O P \\ w_P & w_P & -w_P P \\ \hline 1 & 1 & -P \end{array}\right]. \tag{52}$$

To loopshape with L, we calculate N^L from G using (18):

$$N^L = \begin{bmatrix} G_{11} & G_{12}P^{-1} \\ G_{21} & G_{22}P^{-1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & w_O \\ w_P & w_P & -w_P \\ 1 & 1 & -1 \end{bmatrix}. \tag{53}$$

5.1.1 Sufficient Bounds on L

Theorem 3.1 gives the sufficient upper bound on L for robust performance to be achieved. The upper bound is c_L^{su}, where c_L^{su} at each frequency solves

$$\mu_{\left[\Delta \; \Delta_L\right]} \begin{bmatrix} 0 & 0 & w_O \\ w_P & w_P & -w_P \\ c_L^{su} & c_L^{su} & -c_L^{su} \end{bmatrix} = 1. \tag{54}$$

Δ_L has the same structure as L, namely 1×1, so diag$\{\Delta, \Delta_L\}$ = diag$\{\Delta_O, \Delta_P, \Delta_L\}$ is a 3×3 diagonal matrix with independent blocks.

Theorem 3.1 gives us that $\mu_{\Delta}(M) < 1$ for all frequencies for which $|L|$ is lower than c_L^{su}. An analytical expression can be derived for μ in (54) from the definition of μ. Using this expression, c_L^{su} can be solved analytically. We find that for each frequency

$$\mu_{\Delta}(M) < 1 \Longleftarrow |L| < c_L^{su} = \frac{1-|w_P|}{1+|w_O|}. \tag{55}$$

Similarly, the sufficient lower bound theorem gives

$$\mu_{\Delta}(M) < 1 \Longleftarrow |L| > c_L^{sl} = \frac{1+|w_P|}{1-|w_O|}. \tag{56}$$

The expressions for c_L^{su} and c_L^{sl} above are most easily derived from

$$\mu_{\Delta}(M) = \mu_{\left[\Delta_O \; \Delta_P\right]} \begin{bmatrix} w_O H & w_O H \\ w_P S & w_P S \end{bmatrix} = |w_O H| + |w_P S| = \left|w_O \frac{L}{1+L}\right| + \left|w_P \frac{1}{1+L}\right| \tag{57}$$

combined with the triangle inequality [e.g., use $|1-|L|| \leq |1+L| \leq 1+|L|$ (see [9] for details)].

Notice that the sufficient upper bound on $|L|$ is defined only for $|w_P| < 1$ (typically true for high frequencies), and the sufficient lower bound on $|L|$ is defined only for $|w_O| < 1$ (typically true for low frequencies). These conditions correspond to the requirement that $\mu_{\mathbf{\Delta}}(N_{11}) < 1$ holds for the sufficient upper bound to be defined in Thm. 3.1, and to the requirement that $e(\infty) < k$ holds for the sufficient lower bound to be defined in Thm. 3.4.

5.1.2 Necessary Bounds on L

For the necessary bounds to exist in Theorems 3.8 and 3.11, we need

$$\min_{\Delta_L} \mu_{\begin{bmatrix} \Delta_O & \\ & \Delta_P \end{bmatrix}} \left(F_l \left(\begin{bmatrix} 0 & 0 & w_O \\ w_P & w_P & -w_P \\ 1 & 1 & -1 \end{bmatrix}, \Delta_L \right) \right) < 1. \tag{58}$$

The above minimization can be solved analytically to give

$$\min_{\Delta_L} \mu_{\begin{bmatrix} \Delta_O & \\ & \Delta_P \end{bmatrix}} \left(F_l \left(\begin{bmatrix} 0 & 0 & w_O \\ w_P & w_P & -w_P \\ 1 & 1 & -1 \end{bmatrix}, \Delta_L \right) \right) = \min \{|w_O|, |w_P|\}. \tag{59}$$

This equation is most easily derived using (57). The left hand minimization in (59) is achieved for $\Delta_L^{-1} \to 0$ when $|w_O| < |w_P|$ and for $\Delta_L = 0$ when $|w_O| \geq |w_P|$.

A necessary condition for robust performance is that for each frequency either $|w_O|$ or $|w_P|$ is less than one. This will be assumed in the following derivation of the necessary bounds, since if this condition is not met, then robust performance cannot be met by any controller.

Necessary Upper Bound on L Theorem 3.8 gives the tightest necessary upper bound on L for robust performance to be achieved. The upper bound is c_L^{nu}, where c_L^{nu} at each frequency solves

$$\min_{\|\Delta_L\| \geq c_L^{nu}} \mu_{\begin{bmatrix} \Delta_O & \\ & \Delta_P \end{bmatrix}} \left(F_l \left(\begin{bmatrix} 0 & 0 & w_O \\ w_P & w_P & -w_P \\ 1 & 1 & -1 \end{bmatrix}, \Delta_L \right) \right) = 1. \tag{60}$$

Theorem 3.8 gives us that for $\mu_{\mathbf{\Delta}}(M)$ to be less than 1 it is required that $|L|$ is lower than c_L^{nu}. The minimization in (60) can be solved analytically to give the following expression for c_L^{nu}:

$$\mu_{\mathbf{\Delta}}(M) < 1 \Longrightarrow |L| < c_L^{nu} = \frac{1 - |w_P|}{|w_O| - 1}. \tag{61}$$

Again the expression for c_L^{nu} is most easily derived from (57) combined with the triangle inequality.

Necessary Lower Bound on L Theorem 3.11 gives the tightest necessary lower bound on L for robust performance to be achieved. The upper bound is c_L^{nl}, where c_L^{nl} at each frequency solves

$$\min_{\|\Delta_L\| \leq c_L^{nl}} \mu_{\begin{bmatrix} \Delta_O & \\ & \Delta_P \end{bmatrix}} \left(F_l \left(\begin{bmatrix} 0 & 0 & w_O \\ w_P & w_P & -w_P \\ 1 & 1 & -1 \end{bmatrix}, \Delta_L \right) \right) = 1. \tag{62}$$

Theorem 3.11 gives us that for $\mu_{\mathbf{\Delta}}(M)$ to be less than 1 it is required that $|L|$ is larger than c_L^{nl}. The minimization in (62) can be solved analytically to give the following expression for c_L^{nl}:

$$\mu_{\mathbf{\Delta}}(M) < 1 \Longrightarrow |L| > c_L^{nl} = \frac{|w_P| - 1}{1 - |w_O|}. \tag{63}$$

Again the expression for c_L^{nl} is most easily derived from (57) combined with the triangle inequality.

Notice that the necessary upper bound on $|L|$ is defined only for $|w_O| > 1$ (typically true for high frequencies), and the necessary lower bound on $|L|$ is defined only for $|w_P| > 1$ (typically true for low frequencies). These conditions correspond to the requirement that $g(\infty) > 1$ holds for the necessary upper bound to be defined in Thm. 3.8, and to the requirement that $\mu_{\boldsymbol{\Delta}}(N_{11}) > 1$ holds for the necessary lower bound to be defined in Thm. 3.11.

Remark 5.1 *All the above bounds are derived in [9] using the triangle inequality. The condition required for the necessary bounds to exist,* $\min\{|w_O|, |w_P|\} < 1$, *is the second "algebraic constraint" in [9]. The advantage of Theorems 3.1, 3.4, 3.8, and 3.11 is that the bounds can be calculated for arbitrary uncertainty descriptions including mixed real and complex, and for the design of fully-decentralized controllers.*

5.2 Example 2a—DC Motor with Negligible Viscous Damping

Description Assume the nominal transfer function is the double integrator

$$P(s) = \frac{1}{s^2}. \tag{64}$$

This could describe a DC motor with negligible viscous damping. The nominal model, uncertainty description, and performance specifications for this example come from Braatz et al. [5].

We are interested in good tracking over a bandwidth of about 1. If $|S| < 1/|w_P|$, where

$$w_P = \frac{10}{s^3 + 2s^2 + 2s + 1}, \tag{65}$$

then the tracking error is at most 10% over the desired closed loop bandwidth. The true plant is assumed to have a time-delay, which was covered by a multiplicative uncertainty of magnitude $|w_O|$ in [9], where

$$w_O = \frac{0.21s}{0.1s + 1}. \tag{66}$$

This performance and uncertainty description is that of Example 1a; analytical expressions for the necessary bounds and the sufficient bounds are given there.

Loopshaping Design for Robust Performance Braatz et al. [5] design robust controllers for this plant via loopshaping the closed loop transfer functions—the sensitivity and the complementary sensitivity. Nominal stability was satisfied using the interpolation conditions (50). By combining sufficient bounds on H and S over different frequency ranges, robust performance was guaranteed by the resulting design. The robust controller designed in [5] with the smallest value for μ was given by

$$K = \frac{-\frac{15}{16}s^2 + \frac{1}{2}s + 1}{(0.01s + 1)^2}. \tag{67}$$

Fig. 6 is a plot of the structured singular value for robust performance using this design.

The structure of this controller designed by specifying closed loop transfer functions is somewhat awkward, with a right half plane zero at $s = 4/3$. When loopshaping an open loop transfer function, we can directly specify the structure of the controller. We could try to design a PID controller, but it is clear that the integral term is not needed and would add additional phase lag which would be difficult to counteract using the derivative term. Thus we will design a *PD* controller by loopshaping L.

The formula for a PD controller, where the derivative action is assumed to be effective over one decade, is

$$K = k\frac{\tau_D s + 1}{0.1\tau_D s + 1}, \tag{68}$$

where k is the gain and τ_D is the derivative time.

The loopshaping bounds on L are given in Fig. 7. The open loop transfer function for an example design ($k = 10,\ \tau_D = 0.5$) is also shown. Dropping the second order term in s in the numerator of K in (67)

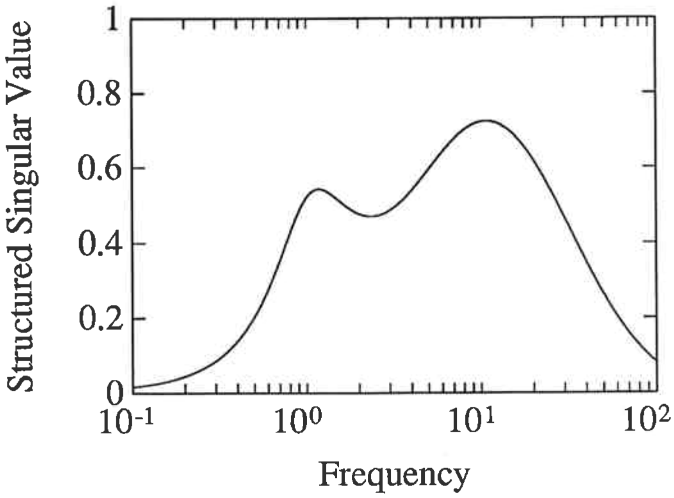

Figure 6: Structured singular value plot for controller in [5] designed via loopshaping closed loop transfer functions.

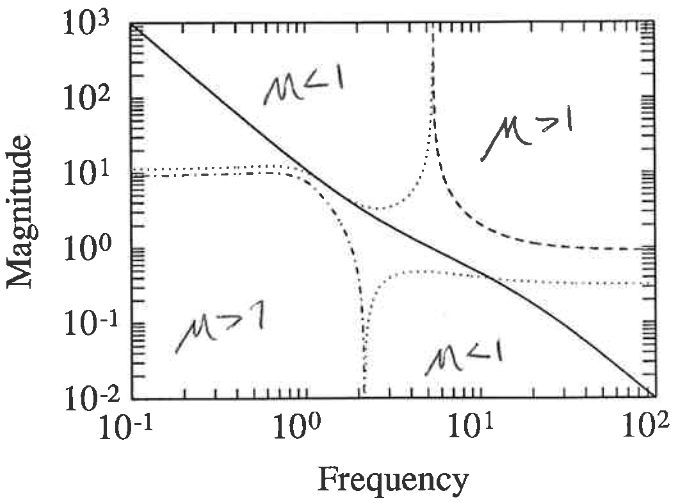

Figure 7: Loopshaping bounds on L for Example 2a. The solid line is $|L|$, the dotted line at low frequencies is the sufficient lower bound, the dotted line at high frequencies is the sufficent upper bound, the dashed line is the necessary upper bound, the dashed-dotted line is the necessary lower bound.

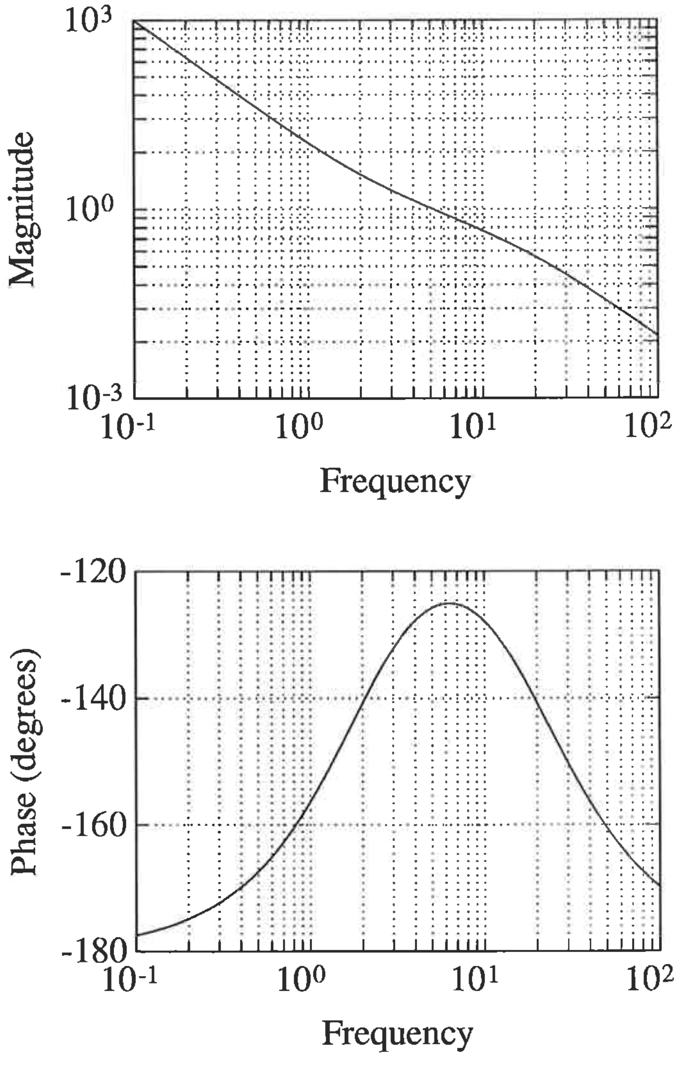

Figure 8: Bode magnitude and phase plots for the open loop transfer function L.

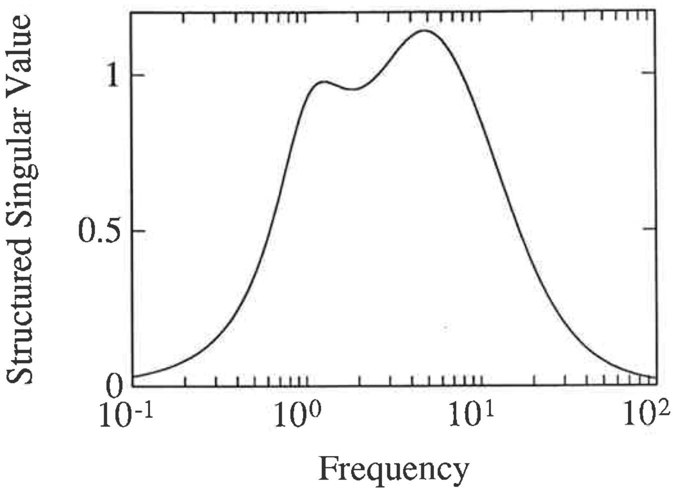

Figure 9: Structured singular value plot to test for robust performance for the controller designed in Example 2a.

gives us a numerator time constant of 0.5—we take this to be the derivative time τ_D. The gain was then chosen to ensure that the loopshape L would satisfy the low and high frequency necessary conditions (this gave $k = 10$). The Bode magnitude and phase plots for the resulting loopshape are given in Fig. 8.

We see from Fig. 8 that the derivative term adds phase lead to the loopshape, which is needed to stabilize the system since the plant has phase lag of 180°. We see from Fig. 7 that the sufficient bounds are satisfied for high and low frequencies. It is impossible to satisfy the sufficient bounds on L at crossover—this will be true in general when loopshaping with an open loop transfer function. Also, though nominal stability cannot be guaranteed *a priori* while loopshaping L, it was still easy to stabilize the system by adding the necessary phase lead at crossover. The closed loop poles with the above PD controller are $\{-13.8, -3.10 \pm 2.21i\}$.

Fig. 9 is a plot of the structured singular value for the PD controller. The maximum μ is equal to 1.14, which is larger than for the previous design (67).

The optimal PD controller was found through optimization to have parameters

$$k = 8.395, \qquad \tau_D = 0.5594, \tag{69}$$

which gives a μ value of 1.10. We see that the PD controller designed via loopshaping L is very close to optimal.

A natural question to ask is why the μ value is so much larger for the PD controller than for the controller designed via closed loop loopshaping in (67). One reason is that the controller in (67) has a much faster time constant (0.01 vs. 0.05). Another reason is that the PD controller stabilizes the system by introducing only phase *lead* at crossover (no RHP zeros), and since the derivative time is active over only one decade, the PD controller can introduce only a limited amount of phase lead at crossover. It is interesting to note that the optimal PD controller which has derivative time active over two decades (and so has a faster time constant and can introduce more phase lead) has a peak μ value of 0.954, which satisfies robust performance.

It is interesting to consider the high and low frequency limits for the necessary and the sufficient bounds.

The necessary and the sufficient upper bounds exist only at high frequencies; this tends to always be the case. The necessary upper bound at high frequencies requires that L roll off sufficiently fast. Since $L = PK$ is strictly proper, L must approach 0 at high frequencies. Note that when $|w_O| >> 1 > |w_P|$ the upper bounds coincide—this coincident bound is then a ***necessary and sufficient*** upper bound for robust performance. In this case, the upper bounds (55,61) both approach

$$\frac{1 - |w_P|}{|w_O|}. \tag{70}$$

This would be expected to hold only at high frequencies because here the uncertainty is largest and the performance requirements are small. The upper bounds do not coincide for high frequencies in Fig. 7 because the uncertainty weight is very lenient for this example $[w_O(\infty) = 2.1]$.

The necessary and the sufficient lower bounds exist only at low frequencies. The necessary lower bound requires that L have sufficiently high gain at low frequencies. Since L has a double integrator, L must approach ∞ at low frequencies. The lower bounds coincide when $|w_P| >> 1 > |w_O|$. In this case, the lower bounds (56,63) both approach

$$\frac{|w_P|}{1-|w_O|}. \tag{71}$$

$|w_P| >> 1 > |w_O|$ usually holds at low frequencies because here the performance requirements are large (for example, for integral control $|w_P|$ approaches infinity as s approaches zero) and the uncertainty is small. As expected, the lower bounds nearly coincide for low frequencies in Fig. 7. The bounds would overlap at low frequencies if integral action had been an explicit performance requirement. Since the plant is a double integrator, integral action is satisfied automatically.

From Fig. 7 we see that we have $|L| >> c_L^{sl}$ at low frequencies and $|L| << c_L^{su}$ at high frequencies. This suggests that the closed loop system can meet a more stringent performance specification at low frequencies and be robust to more uncertainty at high frequencies. We see from Fig. 9 that the structured singular value is much less than 1 at low and high frequencies, which confirms our judgment that the performance and stability requirements are lenient at low and high frequencies. An increase in the steady-state performance and the high frequency uncertainty requirements would lead to a "flatter" structured singular value plot.

6 Gain and Phase Margins for SISO Plants

Gain and phase margin goals are easy to quantify in terms of the sensitivity. The proofs of the following lemmas are left to the reader.

Lemma 6.1 *The sensitivity and gain margin (GM) are related by*

$$|S(j\omega_{gm})| = \frac{GM}{GM-1}, \tag{72}$$

where ω_{gm} is the frequency of the leftmost intersection of $|L(j\omega)|$ with the negative real axis in the SISO Nyquist plot.

Lemma 6.2 *The sensitivity and the phase margin (PM) are related by*

$$|S(j\omega_{pm})| = \frac{1}{\sqrt{2(1-\cos(PM))}}, \tag{73}$$

where ω_{pm} is the frequency of the leftmost intersection of $|L(j\omega)|$ with the unit circle in the SISO Nyquist plot.

The frequencies ω_{gm} and ω_{pm} and commonly referred to as the gain and phase crossover frequencies [12].

To make sure that specified gain and phase margins are met, the sensitivity is shaped to be less than the values given by right hand sides in (72) and (73). The loopshaping bounds (and perhaps a design iteration) suggest where the gain and phase crossovers will be. Below we give an example illustrating this technique. But first we must review the loopshaping bounds on S and H for an SISO plant.

6.1 Example 1b—Loopshaping S and H for an SISO plant

The complete set of loopshaping bounds on the sensitivity and complementary sensitivity were derived by Braatz et al. [5]. These bounds are reviewed here because they will be needed in Example 2b.

Sufficient Upper Bounds on Sensitivity and Complementary Sensitivity The sufficient upper bounds for $|H|$ and $|S|$ are

$$\mu_{\Delta}(M) < 1 \Longleftarrow |H| < c_H^{su} = \frac{1 - |w_P|}{|w_O| + |w_P|} \quad \text{if } |w_P| < 1. \tag{74}$$

$$\mu_{\Delta}(M) < 1 \Longleftarrow |S| < c_S^{su} = \frac{1 - |w_O|}{|w_O| + |w_P|} \quad \text{if } |w_O| < 1. \tag{75}$$

The sufficient lower bound does not exist for closed loop transfer functions.

Necessary Bounds on Sensitivity and Complementary Sensitivity We show in [5] that for the necessary bounds to exist we need

$$\min\{|w_O|, |w_P|\} < 1 \quad \forall\omega. \tag{76}$$

This will be assumed in the following derivation of the necessary bounds, since if this condition is not met, then robust performance cannot be met by any controller.

The necessary upper bounds for $|H|$ and $|S|$ are [5]

$$\mu_{\Delta}(M) < 1 \Longrightarrow |H| < c_H^{nu} = \begin{cases} \frac{1-|w_P|}{|w_O|-|w_P|} & \text{if } |w_O| > 1, \\ \frac{1+|w_P|}{|w_O|+|w_P|} & \text{if } |w_O| \leq 1. \end{cases} \tag{77}$$

$$\mu_{\Delta}(M) < 1 \Longrightarrow |S| < c_S^{nu} = \begin{cases} \frac{1-|w_O|}{|w_P|-|w_O|} & \text{if } |w_P| > 1, \\ \frac{1+|w_O|}{|w_P|+|w_O|} & \text{if } |w_P| \leq 1. \end{cases} \tag{78}$$

Note that the necessary upper bound is defined for all frequencies provided that $\min\{|w_O|, |w_P|\} < 1$.

The necessary lower bounds for $|H|$ and $|S|$ are

$$\mu_{\Delta}(M) < 1 \Longrightarrow |H| > c_H^{nl} = \frac{1 - |w_P|}{|w_O| - |w_P|} \quad \text{if } |w_P| > 1. \tag{79}$$

$$\mu_{\Delta}(M) < 1 \Longrightarrow |S| > c_S^{nl} = \frac{1 - |w_O|}{|w_P| - |w_O|} \quad \text{if } |w_O| > 1. \tag{80}$$

6.2 Example 2b—DC Motor with Time Delay and Negligible Viscous Damping

Description Assume the nominal transfer function is

$$P(s) = \frac{1}{s^2} \cdot \frac{20 - s}{20 + s}. \tag{81}$$

This could describe a DC motor with negligible viscous damping and a time delay of 0.1 s. The time delay is modelled with a first order Padé approximation. The uncertainty description covers the error introduced by this approximation.

The uncertainty and performance specifications are the same as in Example 2a, except that margin specifications must also be met—the gain margin must be greater than 3, and the phase margin must be greater than 45°. From (72) and (73) we see that specifying these margins is equivalent to specifying that $|S(j\omega_{gm})| < 1.5$ and $|S(j\omega_{pm})| < 1.31$.

Loopshaping Design for Robust Performance In Example 1b we listed the bounds for S and H. The upper plot in Fig. 10 gives the loopshaping bounds on H and the lower plot gives the bounds on S. The complementary sensitivity H and sensitivity S are shown for an example design.

Our design approach is to find an S that satisfies nominal stability and has the sufficient bound on S satisfied for one part of the frequency range and the sufficent bound on H satisfied for the other part of the frequency range so that robust performance is guaranteed for all frequencies. This procedure has been described in detail by Braatz et al. [5].

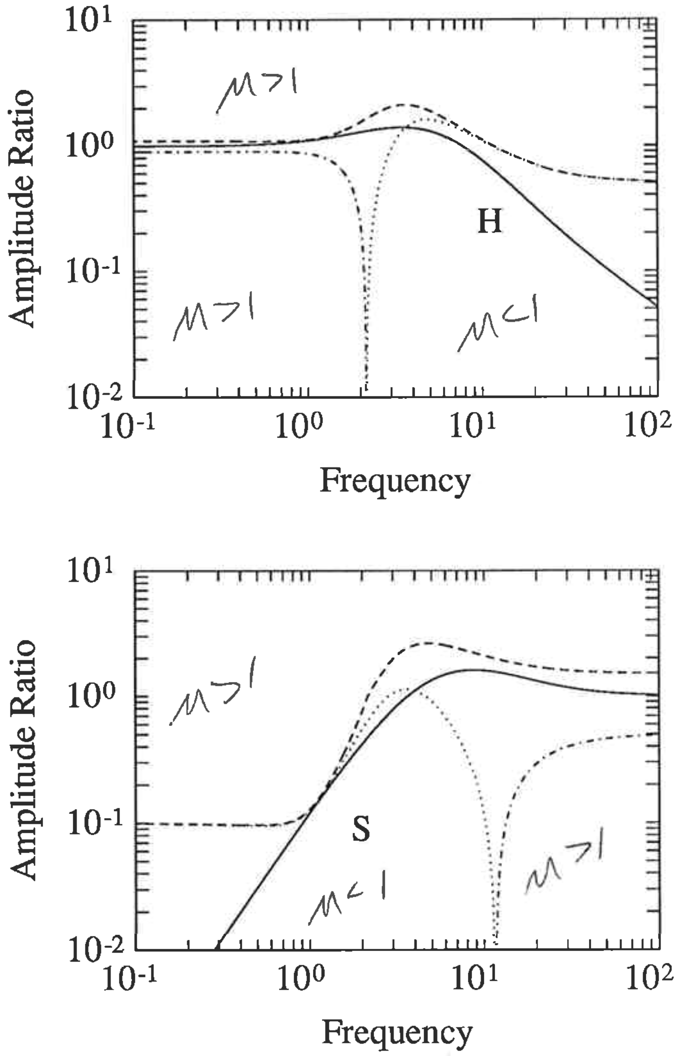

Figure 10: Loopshaping bounds on H and S for the first design in Example 2b. The upper plot is for H and the lower plot is for S. The dashed lines are necessary upper bounds, the dashed and dotted lines are necessary lower bounds, and the dotted lines are sufficient upper bounds.

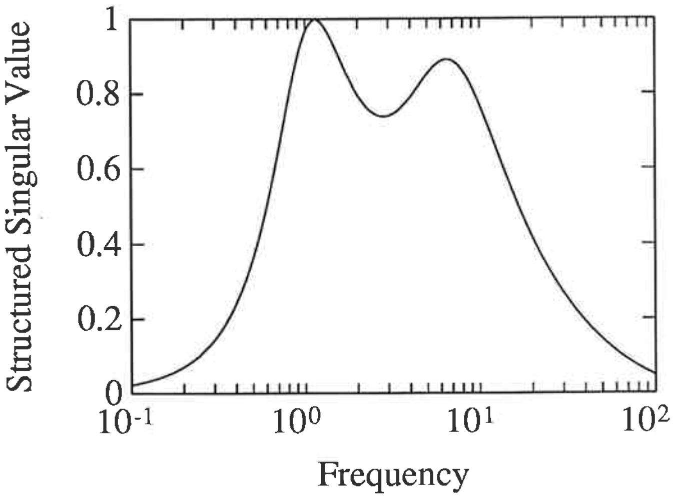

Figure 11: Structured singular value plot to test for robust performance for the first design in Example 2b.

To have internal stability, the two plant poles at $s = 0$ and the plant zero at $s = 20$ cannot be canceled by the controller. So for nominal internal stability, $S = (1 + PK)^{-1}$ must have two zeros at $s = 0$ (interpolation condition (50)), and must satisfy $S(20) = 1$ (interpolation condition (49)). Since the plant is strictly proper and the controller must be proper, S must also satisfy $S(\infty) = 1$.

Let us try the following form for S which gives a nominally stable system:

$$S = \frac{\lambda^2 s^2}{(\lambda s + 1)^2} \cdot \frac{s + a}{s + b}, \tag{82}$$

where

$$a = \frac{(20\lambda + 1)^2}{(20\lambda)^2}(20 + b) - 20, \tag{83}$$

and b is arbitrary. For simplicity, we initially take $b = 1/\lambda$ so that the denominator time constants of S are equal. The complementary sensitivity H is given by $H = 1 - S$. The controller calculated from $K = (SP)^{-1}(1 - S)$ is improper, and so is augmented with the second order filter

$$\frac{1}{(0.01s + 1)^2} \tag{84}$$

before calculating gain and phase margins and the structured singular value. The closed loop poles are calculated to ensure that nominal stability is still satisfied by the augmented controller.

The loopshaping bounds in Fig. 10 suggest that we try $\lambda \approx 0.2$. Plotting S and H for different values of λ near 0.2 shows that the necessary bounds on S and H are satisfied for $0.16 < \lambda < 0.18$. The design shown in Fig. 10 is for $\lambda = 0.18$. Since the sufficient bound on S is satisfied for $\omega < 3.8$ and the sufficient bound on H is satisfied for $\omega > 4.0$, we expect robust performance to be approximately satisfied. The structured singular value is plotted in Fig. 11, and the peak μ value is 0.998.

We calculated the margins for several controllers with λ in the range from 0.16 to 0.18. We found that $\lambda = 0.18$ gives the best margins of all the controllers that satisfy robust performance and are given by (82), (83), and $b = 1/\lambda$. The margins, with their respective crossover frequencies, are

$$GM = 2.93 < 3, \qquad \omega_{gm} = 13.9,$$
$$PM = 38.7^o < 45^o, \qquad \omega_{pm} = 4.93.$$

The margin specifications are not satisfied.

It is easy to modify the design to improve the margins. The peak in the sensitivity occurs at $\omega = 7.83$, which is between ω_{pm} and ω_{gm}. This suggests that we should be able to improve the margins by reducing

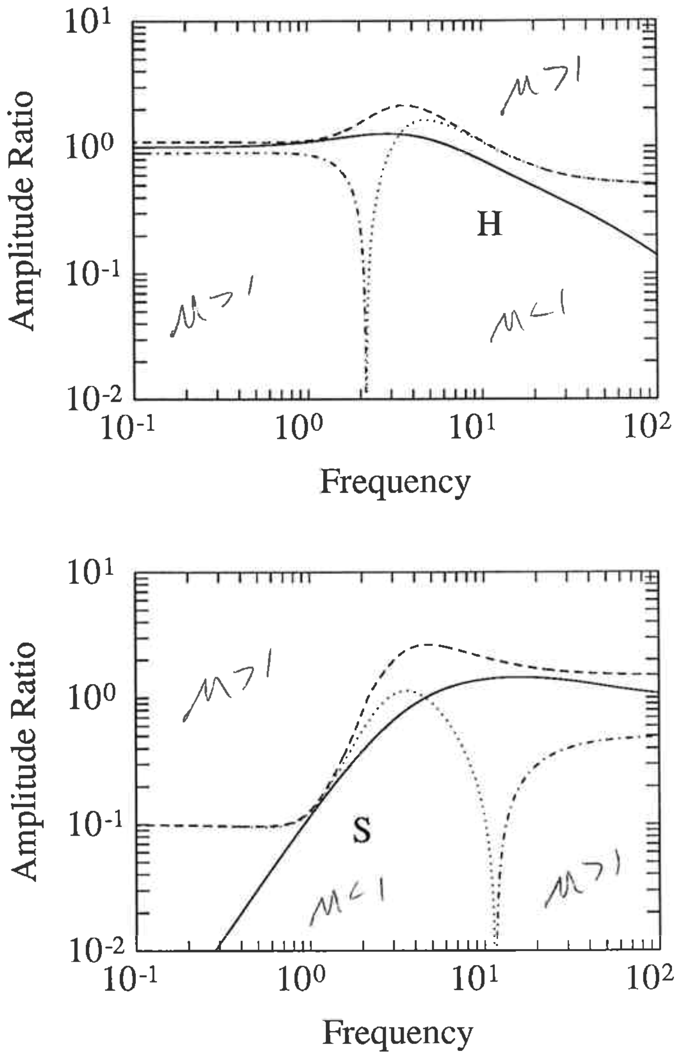

Figure 12: Loopshaping bounds on H and S for the second design in Example 2b. The upper plot is for H and the lower plot is for S. The dashed lines are necessary upper bounds, the dashed-dotted lines are necessary lower bounds, and the dotted lines are sufficient upper bounds.

the peak in the sensitivity. This can be done by choosing $b = 10/\lambda$. Again we augment the controller with the second order filter (84) before calculating margins and the structured singular value.

The loopshaping bounds are given for $\lambda = 0.28$ in Fig. 12. The necessary bounds on S and H are satisfied for all frequencies. Since the sufficient bound on S is satisfied for $\omega < 4.4$ and the sufficient bound on H is satisfied for $\omega > 3.5$, μ must be less than 1 for all frequencies. The structured singular value is plotted in Fig. 13. The peak μ value of 0.96 is less than 1, as implied by the satisfaction of the sufficient bound on S and/or H for each frequency. The closed loop poles are $\{-116.1 \pm 44.5i, -20.0, -15.5, -6.92, -2.75\}$, so the system is nominally stable.

The margins for this design are

$$GM = 3.02 > 3, \qquad \omega_{gm} = 19.6,$$
$$PM = 47^o > 45^o, \qquad \omega_{pm} = 5.42.$$

Both the crossover frequencies and the peak in the sensitivity is shifted to higher frequencies. The peak in the sensitivity is reduced from 1.85 to 1.65, and this results in the improved margins.

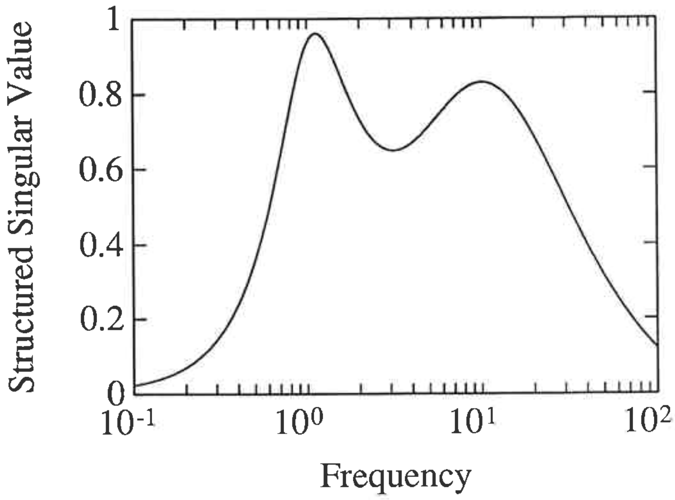

Figure 13: Structured singular value plot to test for robust performance for the second design in Example 2b.

7 Multiple Performance Specifications

Robust loopshaping is easily extended to design controllers to meet multiple performance specifications. For example, one might want a controller that remains stable under slow sensor drift or variations in actuator or sensor gain (this is referred to as *fault tolerance*). If there are sensors or actuators that are prone to failures, then we would like to specify that the closed loop system remain stable or satisfy some minimum performance whenever these sensors/actuators fail (this is referred to as *failure tolerance*). Also, often specifications are in terms of nominal performance and robust stability, instead of robust performance. Multiple performance specifications are easy to handle using loopshaping—the bounds are calculated individually for each specification, and the most restrictive bounds are used for loopshaping.

7.1 Nominal Performance + Robust Stability

Often the designer prefers to give specifications not in terms of robust performance, but in terms of nominal performance plus robust stability. For example, the specifications for the 1990-92 ACC Benchmark Problem [3, 4] are that the overshoot and settling time should be minimized for the nominal plant, and that stability is satisfied for some set of plants. Separate specifications are used whenever the designer expects that an overall robust performance specification will lead to an overly conservative design.

7.2 Fault Tolerance

Fault tolerance refers to the ability of the control system to meet some performance specifications even when actuators and sensors become faulty. It is easy to include fault tolerance specifications through an additional μ condition. Below we show how to do this for the commonly occuring faults of gain variation and slow drift.

Gain Variation To develop a system that maintains a given performance even under gain variation in the actuators or the sensors, just treat the gain variation as real parametric uncertainty.

Below we show how to treat actuator gain variations for two cases: 1) without additional uncertainty, and 2) with additional uncertainty. A similar development can be done for sensor gain variations or for combined variations in actuator and sensor gains. For stability of fully-decentralized control systems without additional uncertainty, sensor gain variations are equivalent to actuator gain variations.

The nominal controller is defined to be $K(s)$. Then the controller with gain variation can be described by $\hat{K}(s) = EK(s)$, where $E = \text{diag}\{\epsilon_i\}$, and $\epsilon_{i,low} \leq \epsilon_i \leq \epsilon_{i,high}$. We can write the set of E described by the gain variation as $E = \bar{E} + W_r \Delta^r$, where $\bar{E} = \text{diag}\{\bar{\epsilon}_i\}$, $W_r = \text{diag}\{w_i\}$,

$$\bar{\epsilon}_i = \frac{\epsilon_{i,high} + \epsilon_{i,low}}{2}, \tag{85}$$

$$w_i = \frac{\epsilon_{i,high} - \epsilon_{i,low}}{2}, \tag{86}$$

and Δ^r is a diagonal Δ-block with real independent uncertainties.

Standard block diagram manipulations are used to arrive at the $M - \Delta$ block structure in Fig. 1, where $\Delta = \Delta^r$ and

$$M = -(I + K(s)P(s)\bar{E})^{-1}K(s)P(s)W_r. \tag{87}$$

Stability is obtained for all variations in gain if and only if $\mu_{\Delta^r}(M) < 1$.

To design such controllers via loopshaping, we need to have the expression for the G matrix in Fig. 4. This matrix is

$$G = \begin{bmatrix} 0 & I \\ -PW_r & -P\bar{E} \end{bmatrix}. \tag{88}$$

The N^T matrices needed for calculating loopshaping bounds are determined using (16)-(22).

If we are interested in maintaining stability or performance with respect to other perturbations, then the expressions for M and G are somewhat more complicated. The designer should avoid asking for the full performance under large variations in actuator/sensor gains; otherwise the designed controller will be conservative, i.e. will perform sluggishly even when the actuators and sensors behave perfectly. Let the original system be described by $\hat{G}(s)$ with uncertainty $\hat{\Delta}$.

The new Δ matrix is $\Delta = \text{diag}\{\hat{\Delta}, \Delta^r\}$. The new M matrix is

$$M = \begin{bmatrix} \hat{G}_{11} + \hat{G}_{12}\bar{E}K(I - \hat{G}_{22}\bar{E}K)^{-1}\hat{G}_{21} & \hat{G}_{12}(I + \bar{E}K(I - \hat{G}_{22}\bar{E}K)^{-1}\hat{G}_{22})W_r \\ K(I - \hat{G}_{22}\bar{E}K)^{-1}\hat{G}_{21} & K(I - \hat{G}_{22}\bar{E}K)^{-1}\hat{G}_{22}W_r \end{bmatrix}. \tag{89}$$

The new G matrix is

$$G = \begin{bmatrix} \hat{G}_{11} & \hat{G}_{12}W_r & \hat{G}_{12}\bar{E} \\ 0 & 0 & I \\ \hat{G}_{21} & \hat{G}_{22}W_r & \hat{G}_{22}\bar{E} \end{bmatrix}. \tag{90}$$

Slow Drift It is quite common for a sensor reading to slowly drift. This slow drift does not affect closed loop stability (provided the measurement sensitivity [gain] is unchanged), and can be treated as a slow disturbance at the output of the plant that must be rejected by the controller. This is included as an additional specification in defining robust performance. The disturbance weight is chosen to have higher gain at low frequency and a time constant approximately equal to the time constant of the sensor drift. For example, the disturbance weight could be chosen to be

$$w_d(s) = M\frac{(\tau_{drift}/10)s + 1}{\tau_{drift}s + 1}, \tag{91}$$

where M is the magnitude and τ_{drift} is the time constant of the sensor drift. Fig. 14 is a bode magnitude plot of the disturbance weight for $M = 0.2$ and $\tau_{drift} = 10$.

Another reasonable choice for the disturbance weight is the integrator

$$w_d(s) = \frac{M}{s}. \tag{92}$$

7.3 Failure Tolerance

Conventional feedback control designs for a MIMO plant may result in poor performance, or even instability, in the event of actuator or sensor outages, even though it may be possible to control the plant using only the surviving inputs and outputs. *Failure tolerance* refers to the ability of the control system to meet some

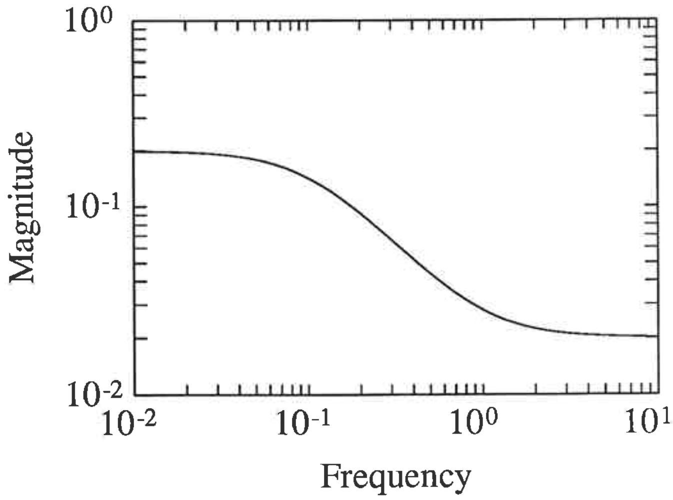

Figure 14: A disturbance weight to describe slow sensor drift.

(weaker than normal) performance specifications even though a prespecified set of actuators and sensors fail. Typically we will design the control system to be failure tolerant to only those actuators and sensors which we suspect might fail—otherwise the designed controller could be overly conservative.

For an example illustrating the importance of designing failure tolerant controllers, consider a distillation column where the setpoints are the top and bottom compositions. Composition measurements are often too slow for effective control, so usually the controller is designed to use temperature measurements. The drawback of using temperature measurements only is that it is then impossible to have zero steady-state error in the compositions. Thus it is advantageous to design the control system that uses temperature measurements, and also uses composition measurements when these are available. Composition analyzers are typically prone to failure. When a composition measurement fails, the control system should be capable of giving acceptable performance using only temperature measurements. Such a controller is said to be failure tolerant.

Even though the importance of designing failure tolerant controllers is clear, relatively few design methods have been proposed (see Veillette et al [21] for a survey). None of these methods, except for the method of Veillette et al [21], gives guarantees on system performance. The method of Veillette et al [21] designs for nominal performance, i.e. the satisfaction of a bound on the H_∞-norm of some transfer function of interest. The robust loopshaping method is the only method which designs failure tolerant controllers for *robust* performance.

The first step in designing a failure tolerant control system is to specify which sensor/actuator combinations are expected to fail. Then a performance specification is chosen for each set of sensor/actuator failures. Sometimes the requirement on the failed system is only that the closed loop remains stable. Once the different performance specifications are set, then robust loopshaping bounds can be calculated for each separate μ-problem and the most restrictive robust loopshaping bounds are used to design the controller. This approach will be illustrated in Example 3.

Below we discuss a very strong notion of failure tolerance in which closed loop stability is required for any combination of actuator failures. We then extend this notion to uncertain systems. Similar definitions can be given for sensor failures.

Integrity *Integrity* is defined by Campo and Morari [6].

Definition 7.1 *The closed loop system demonstrates* integrity *if* $\tilde{K}(s) = EK(s)$ *stabilizes* $P(s)$ *for all* $E \in \mathcal{E}_{1/0}$ *where*

$$\mathcal{E}_{1/0} \equiv \{E = \text{diag}(\epsilon_i) \mid \epsilon_i \in \{0,1\}, i = 1, \ldots, n\}. \tag{93}$$

Note that for a system to demonstrate integrity, the plant $P(s)$ must be stable.

A closed loop system which demonstrates integrity remains stable as subsystem controllers are arbitrarily brought in and out of service. Note that integrity does not imply sensor or actuator failure tolerance unless the failure is recognized and the affected control loop taken out of service.

It is clear that whether a system demonstrates integrity can be tested through 2^n stability (eigenvalue) determinations [6].

Robust Integrity We can generalize the definition of integrity to include robustness. *Robust integrity* is defined below.

Definition 7.2 *The closed loop system demonstrates* robust integrity *if the system is stable with* $\tilde{K}(s) = EK(s)$ *for all* $E \in \mathcal{E}_{1/0}$ *and all* $\Delta \in \mathbf{B\Delta}$ *where*

$$\mathcal{E}_{1/0} \equiv \{E = \mathrm{diag}(\epsilon_i) \mid \epsilon_i \in \{0,1\}, i = 1, \ldots, n\}. \tag{94}$$

Note that for a system to demonstrate robust integrity, the plant must be stable under all allowed perturbations. Note also that robust integrity implies integrity.

A closed loop system which demonstrates robust integrity remains robustly stabilized as subsystem controllers are arbitrarily brought in and out of service. Robust integrity does not imply sensor or actuator failure tolerance unless the failure is recognized and the affected control loop taken out of service.

It is clear that whether a system demonstrates robust integrity can be tested through 2^n nominal stability (eigenvalue) and 2^n robust stability (μ) calculations.

7.4 Fault and Failure Tolerance

A very strong notion of fault tolerance was defined by Campo and Morari [6] for fully-decentralized controllers. The requirement is that the closed loop system remains stable under arbitrary detuning of the controller gains. For fully-decentralized control systems, this is equivalent to arbitrary detuning of the actuator/sensor gains.

Decentralized Unconditional Stability The following definition of *decentralized unconditional stability* is slightly modified from that of Campo and Morari [6].

Definition 7.3 *Assume* $K(s)$ *is fully-decentralized. The closed loop system is* decentralized unconditionally stable *(DUS) if* $\tilde{K}(s) = EK(s)$ *stabilizes* $P(s)$ *for all* $E \in \mathcal{E}_D$ *where*

$$\mathcal{E}_D \equiv \{E = \mathrm{diag}(\epsilon_i) \mid \epsilon_i \in (0,1), i = 1, \ldots, n\}. \tag{95}$$

Note that for DUS to make sense the plant $P(s)$ must be stable.

A closed loop system which is DUS remains stable as the gains of each controller subsystem are *independently* detuned. The following result is a computable necessary and sufficient condition for DUS.

Theorem 7.4 *Assume* $K(s)$ *is decentralized. Define* Δ^r *to be a diagonal* Δ*-block with independent real uncertainties. Then the closed loop system is DUS if and only if* $(I + \frac{1}{2}K(s)P(s))^{-1}K(s)P(s)$ *is stable and*

$$\mu_{\Delta^r}(-\frac{1}{2}(I + \frac{1}{2}K(s)P(s))^{-1}K(s)P(s)) \leq 1, \quad \forall \omega. \tag{96}$$

Proof: Let $\bar{E} = W_r = (1/2)I$ in (87). The conditions $\mu_{\Delta^r}(-\frac{1}{2}(I + \frac{1}{2}K(s)P(s))^{-1}K(s)P(s)) \leq 1$, $\forall \omega$ and $(I + \frac{1}{2}K(s)P(s))^{-1}K(s)P(s)$ is stable ensure that the closed loop system is stable for all $\epsilon_i \in (0,1)$. QED.

The closed loop system cannot be DUS when the controller $K(s)$ has poles in the open right half plane—this is because some minimum amount of feedback is required to have closed loop stability.

To calculate loopshaping bounds to meet the μ condition in Thm. 7.4, we need the expression for the $G(s)$ matrix in Fig. 4. This matrix is given by (88) with $\bar{E} = W_r = (1/2)I$:

$$G = \begin{bmatrix} 0 & I \\ -(1/2)P & -(1/2)P \end{bmatrix}. \tag{97}$$

Robust Decentralized Unconditional Stability We can generalize the definition of DUS to include robustness. Clearly with arbitrary detuning of single loop controller gains it is not reasonable to ask for performance of the arbitrarily detuned system to be better than open loop. But it could be reasonable to expect that the system remains robustly stable under arbitrary detuning of single loop controller gains.

Definition 7.5 *Assume $K(s)$ is decentralized. The closed loop system is* robust decentralized unconditionally stable *(RDUS) if the system is stable with $\tilde{K}(s) = EK(s)$ for all $E \in \mathcal{E}_{1/0}$ and all $\Delta \in \mathbf{B\Delta}$ where*

$$\mathcal{E}_D \equiv \{E = \mathrm{diag}(\epsilon_i) \mid \epsilon_i \in (0,1), i = 1, \ldots, n\}. \tag{98}$$

Note that for RDUS to be satisfied, the plant must be stable under all allowed perturbations.

The following result is a computable necessary and sufficient condition for RDUS.

Theorem 7.6 *Assume $K(s)$ is decentralized, and that the uncertain system is described by $\hat{G}(s)$ and $\hat{\Delta}$. Define Δ^r to be a diagonal Δ-block with independent real uncertainties. Then the closed loop system is RDUS if and only if $M(s)$ is stable and*

$$\mu_\Delta(M) \leq 1, \quad \forall\omega, \tag{99}$$

where $\Delta = \mathrm{diag}\{\hat{\Delta}, \Delta^r\}$, and

$$M = \begin{bmatrix} \hat{G}_{11} + \frac{1}{2}\hat{G}_{12}K(I - \frac{1}{2}\hat{G}_{22}K)^{-1}\hat{G}_{21} & \frac{1}{2}\hat{G}_{12}(I + \frac{1}{2}K(I - \frac{1}{2}\hat{G}_{22}K)^{-1}\hat{G}_{22}) \\ K(I - \frac{1}{2}\hat{G}_{22}K)^{-1}\hat{G}_{21} & \frac{1}{2}K(I - \frac{1}{2}\hat{G}_{22}K)^{-1}\hat{G}_{22} \end{bmatrix}. \tag{100}$$

Proof: Let $\bar{E} = W_r = (1/2)I$ in (89). The conditions $\mu_\Delta(M) \leq 1$, $\forall\omega$ and $M(s)$ stable ensure that the closed loop system is stable for all $\epsilon_i \in (0,1)$ and $\hat{\Delta} \in \mathbf{B}\hat{\mathbf{\Delta}}$. QED.

The closed loop system cannot be RDUS when the controller $K(s)$ has poles in the open right half plane—this is because some minimum amount of feedback is required to have closed loop stability.

To calculate loopshaping bounds to meet the μ condition in Thm. 7.6, we need the expression for the $G(s)$ matrix in Fig. 4. This matrix is given by (90) with $\bar{E} = W_r = (1/2)I$:

$$G = \begin{bmatrix} \hat{G}_{11} & (1/2)\hat{G}_{12} & (1/2)\hat{G}_{12} \\ 0 & 0 & I \\ \hat{G}_{21} & (1/2)\hat{G}_{22} & (1/2)\hat{G}_{22} \end{bmatrix}. \tag{101}$$

Remark 7.7 *Actually, the definition of DUS given by Campo and Morari [6] requires that the system is stable for all $\epsilon_i \in [0,1]$—we will refer to this version as* CDUS *(*closed *DUS).*

When $K(s)$ is stable, a necessary and sufficient test for CDUS is given by Thm. 7.4 except with the condition $\mu < 1$ replacing $\mu \leq 1$ in (96).

When $K(s)$ is an integral controller, Nwokah and co-workers [13, 16, 17] claim to derive a computable necessary and sufficient condition for CDUS (though they mistakenly refer to CDUS as Decentralized Integral Controllability), but it is quite simple to show that their results are incorrect. CDUS can be checked through a finite number of stability and μ tests—just use Thm. 7.4 to check the interior of the ϵ-hypercube, and test the boundary[2] (the points, edges, faces, etc.) through additional μ tests. The number of μ-tests required grows rapidly with the size of the system. It is not known if all these μ tests are really needed. The reason CDUS cannot be tested by one μ test is that setting the proportional gain to zero in a controller with integral action will remove the feedback around the integrator, which will then be a limit of instability. This problem is only of a mathematical nature—it does not correspond to a physical problem since the integral action is turned off when the proportional gain is set to zero. A problem formulation that avoids this mathematical artifact is currently being investigated.

When $K(s)$ is integral, the μ test in Thm. 7.4 is a tight necessary condition for CDUS, so as long as μ approaches 1 only at zero frequency and is strictly less than 1 for all other frequencies, then the designer can feel fairly confident that the system is CDUS.

CRDUS can be defined similarly, and a similar discussion applies as for CDUS.

[2] The boundary of the ϵ-hypercube is the set of detuned systems with at least one of the $\epsilon_i = 0$ or 1.

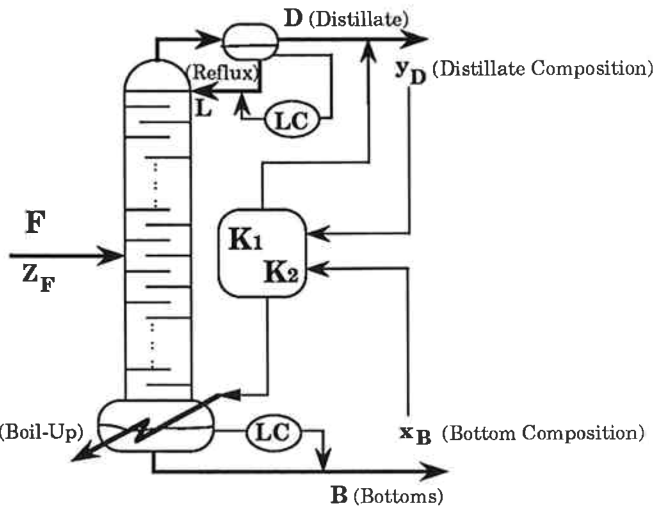

Figure 15: High Purity Distillation Column in DV configuration

Robust Decentralized Detunability Detuning a controller refers to changing some parameter in the controller or in the control synthesis procedure so that the control action becomes less aggressive. For example, in Linear Quadratic Control detuning refers to increasing the magnitude of the control weight. In decentralized Internal Model Control, detuning refers to increasing the IMC filter time constants in each single loop controller [11]. The special case of detuning the single loop controller gains in a decentralized controller was discussed earlier in the sections on DUS and RDUS.

Hovd [11] introduced the following very general definition for *robust decentralized detunability.*

Definition 7.8 For a given design method, *a closed loop system is* robust decentralized detunable *(RDD) if each single loop controller can be detuned independently by an arbitrary amount without endangering robust stability.*

Whenever the controller is detuned by varying parameters in the controller, RDD can be tested via a μ test where the variation in parameters is covered by real uncertainty (the real uncertainty must be *independent* for *arbitrary* detuning). Both the robust performance and the "RDD" loopshaping bounds are plotted and the most restrictive of the bounds are used in the design. The resulting controller meets robust performance and gives a system which is RDD. The procedure is illustrated in Example 3.

In Example 3 we design a decentralized controller via loopshaping $\tilde{H}$ and $\tilde{S}$. *For this design procedure*, the closed loop system is RDD if the time constants for the single loop controllers can be increased independently by an arbitrary amount without endangering robust stability.

7.5 Example 3—Robust Fault/Failure Tolerant Decentralized Controller Design for a High-Purity Distillation Column

7.5.1 Description

For this example we will use the loopshaping bounds to design a robust fault/failure tolerant decentralized controller for the high-purity distillation column given in [19] and discussed in more detail in [20]. The nominal model is:

$$P = \frac{1}{75s+1}\begin{bmatrix} -0.878 & 0.014 \\ -1.082 & -0.014 \end{bmatrix}. \tag{102}$$

This nominal model may correspond to a high-purity distillation column using distillate and boilup as manipulated inputs to control top and bottom composition (see Fig. 15) using measurements of the top and bottom compositions. The uncertainty and performance weights are

$$w_I(s) = 0.1\frac{5s+1}{0.25s+1}, \qquad w_P(s) = 0.25\frac{7s+1}{7s}. \tag{103}$$

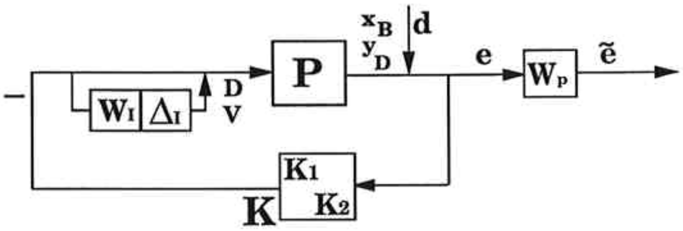

Figure 16: The plant with input uncertainty Δ_I of magnitude $w_I(s)$. Robust performance is satisfied if $\overline{\sigma}\left(w_P(I+\hat{P}K)^{-1}\right) \leq 1$ for all Δ_I with $||\Delta_I||_\infty \leq 1$.

The robust performance condition is a bound on the sensitivity, *i.e.* $\overline{\sigma}(\hat{S}) < 1/|w_P|$, $\forall \hat{P} \in \Pi$. The input uncertainty includes actuator uncertainty and neglected right half plane zeros of the plant. The performance bound implies integral action and a closed loop bandwidth of at least 1/7. The uncertainty block Δ_I is a diagonal 2×2 matrix (independent actuators) and the performance block Δ_P is a full 2×2 matrix. Fig. 16 is a block diagram of the system.

7.5.2 Loopshaping Design For Robust Performance and Fault/Failure Tolerance

Robust performance is satisfied if and only if $\mu_{\mathbf{\Delta}}(M) < 1$ for all frequencies where

$$M = \begin{bmatrix} -w_I P^{-1} H P & -w_I P^{-1} H \\ w_P S P & w_P S \end{bmatrix}, \qquad \Delta = \begin{bmatrix} \Delta_I & \\ & \Delta_P \end{bmatrix}. \tag{104}$$

$N^{\tilde{H}}$ and $N^{\tilde{S}}$ can be found by inspection or G could be found and the equations in Section 2.5 could be used.

$M = F_l(N^{\tilde{H}}, \tilde{H})$, where

$$N^{\tilde{H}}_{11} = \begin{bmatrix} 0 & 0 \\ w_P P & w_P I \end{bmatrix}, N^{\tilde{H}}_{12} = \begin{bmatrix} -w_I \tilde{P}^{-1} \\ -w_P P \tilde{P}^{-1} \end{bmatrix}, N^{\tilde{H}}_{21} = \begin{bmatrix} P & I \end{bmatrix}, N^{\tilde{H}}_{22} = I - P\tilde{P}^{-1}. \tag{105}$$

$M = F_l(N^{\tilde{S}}, \tilde{S})$, where

$$N^{\tilde{S}}_{11} = \begin{bmatrix} -w_I I & -w_I P^{-1} \\ 0 & 0 \end{bmatrix}, N^{\tilde{S}}_{12} = \begin{bmatrix} w_I P^{-1} \\ w_P I \end{bmatrix}, N^{\tilde{S}}_{21} = \begin{bmatrix} \tilde{P} & \tilde{P}P^{-1} \end{bmatrix}, N^{\tilde{S}}_{22} = I - \tilde{P}P^{-1}. \tag{106}$$

Note that the equation (35b) given in [19] is incorrect.

Applying Theorems 3.1, 3.8, and 3.11 with $N^{\tilde{H}}$ gives the sufficient upper bound and the necessary upper and lower bounds on $\tilde{H}$ for robust performance to be achieved. Similarly, applying the theorems using $N^{\tilde{S}}$ gives the corresponding bounds for $\tilde{S}$.

The loopshaping bounds for $\tilde{H} = hI$ and $\tilde{S} = sI$ are plotted in Fig. 17. Braatz et al [5] describe how to use the robust loopshaping bounds to design h and s to meet robust performance. The form for h was chosen to be

$$h = \frac{1}{\lambda s + 1}. \tag{107}$$

Since $s = 1 - h$, we have

$$s = \frac{\lambda s}{\lambda s + 1}. \tag{108}$$

It was found that for $\lambda = 4$ the sufficient bound on h was satisfied for $\omega \geq 0.45$, and the sufficient bound on s was satisfied for $\omega \leq 0.45$. Combining the sufficient conditions over the different frequency ranges ensured that $\mu < 1$ for all frequency (Fig. 18 is the μ plot for robust performance). Nominal stability was shown to be satisfied by calculating the closed loop poles using the full plant P. The controller corresponding to

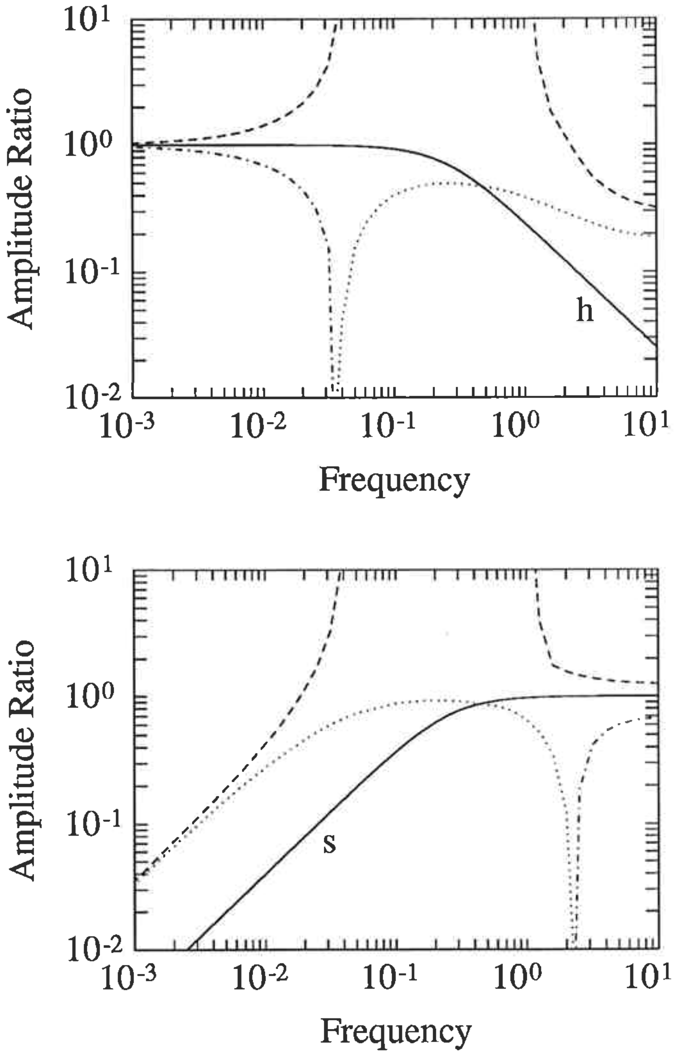

Figure 17: Robust performance loopshaping bounds for Example 3. The upper plot is for h and the lower plot is for s. The dashed lines are necessary upper bounds, the dashed and dotted lines are necessary lower bounds, and the dotted lines are sufficient upper bounds.

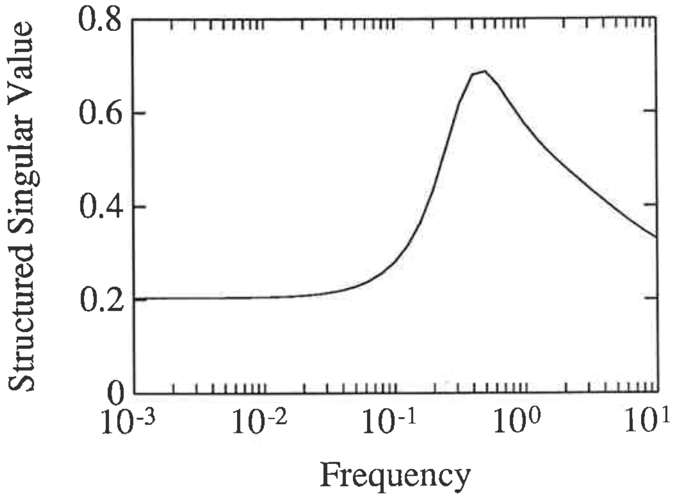

Figure 18: Structured singular value plot to test for robust performance in Example 3.

$\tilde{S} = sI$ and $\tilde{H} = hI$ is

$$K = (\tilde{S}\tilde{P})^{-1}(I - \tilde{S}) = \frac{75s+1}{4s}\begin{bmatrix} -\frac{1}{0.878} & 0 \\ 0 & -\frac{1}{0.014} \end{bmatrix}. \tag{109}$$

Now we want to design a controller that meets some fault/failure tolerance specifications. First we will design the controller so that the closed loop system is RDD. Then we test that the resulting closed loop system demonstates integrity, robust integrity, DUS, and RDUS.

RDD To design for RDD, we plot in Fig. 19 the loopshaping bounds for robust stability where Δ_T is chosen to be a diagonal block with independent elements (the bounds are calculated by applying the Thm. 3.1 on the appropriate submatrices of $N^{\tilde{H}}$ and $N^{\tilde{S}}$ in (105) and (106)).

The closed loop system is RDD if the system remains robustly stable as the controller is dynamically detuned. Dynamic detuning for this example refers to increasing the single loop closed loop time constants λ. A careful consideration of Fig. 19 shows that either the sufficient bound on h or the sufficient bound on s is satisfied for all frequencies for all $\lambda_i \geq 1.8$; thus the system given by $\lambda = 4$ is RDD. A less conservative bound on the λ_i can be derived by directly loopshaping λ (see [11] for details), but deriving the bounds using h and s allows a direct comparison of the robust performance bounds in Fig. 17 and the RDD robust stability bounds in Fig. 19. This comparison shows that the bounds in Fig. 17 are more restrictive, and only these are used to loopshape the controller.

7.5.3 Fault/Failure Tolerance Analysis of Design

We will now test the closed loop system with the designed controller to ensure that is satisfies integrity, robust integrity, DUS, and RDUS.

Integrity The following four transfer functions are stable for $\lambda = 4$:

$$P, \qquad [(\epsilon_1, \epsilon_2) = (0,0)]; \tag{110}$$

$$M_{11} = -w_I P^{-1} H P, \qquad [(\epsilon_1, \epsilon_2) = (1,1)]; \tag{111}$$

$$-w_I K_1 (1 + P_{11} K_1)^{-1} P_{11}, \qquad [(\epsilon_1, \epsilon_2) = (1,0)]; \tag{112}$$

$$-w_I K_2 (1 + P_{22} K_1)^{-1} P_{22}, \qquad [(\epsilon_1, \epsilon_2) = (0,1)]; \tag{113}$$

thus the system has integrity.

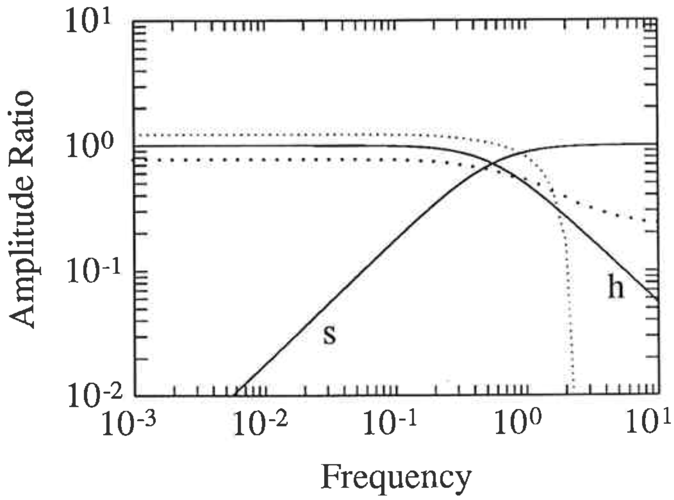

Figure 19: Robust stability loopshaping bounds for Example 3. The solid lines are h and s for $\lambda = 1.8$. The widely spaced dotted line is the sufficient bound for h. The thinly spaced dotted line is the sufficient bound for s.

Robust Integrity To test robust integrity for a 2×2 system, we need to check robust stability for four failure conditions. Nominal stability was tested above (for testing integrity), so we need test only the μ conditions here.

We have robust stability when all loops are turned off provided $P(I + w_I\Delta_I)$ is stable. That $P(I + w_I\Delta_I)$ is stable follows since P, w_I, and Δ_I are stable.

Robust stability for the overall system is satisfied since $\mu_{\Delta_I}(M_{11}) = 0.3 < 1$. Robust stability for the cases when exactly one loop has failed is satisfied since

$$\mu_{\Delta_{I,11}}(-w_I K_1(1 + P_{11}K_1)^{-1}P_{11}) = 0.12 < 1, \qquad [(\epsilon_1, \epsilon_2) = (1, 0)]; \tag{114}$$
$$\mu_{\Delta_{I,22}}(-w_I K_2(1 + P_{22}K_2)^{-1}P_{22}) = 0.12 < 1, \qquad [(\epsilon_1, \epsilon_2) = (0, 1)]. \tag{115}$$
$$\tag{116}$$

Since all four conditions are satisfied, the system demonstrates robust integrity.

DUS, RDUS Since DUS is implied by RDUS, we will only test RDUS here.

The $\hat{G}$ and $\hat{\Delta}$ matrices needed to apply Thm. 7.6 are derived directly from the block diagram in Fig. 16:

$$\hat{G} = \begin{bmatrix} 0 & -w_I I \\ P & -P \end{bmatrix}, \qquad \hat{\Delta} = \Delta_I. \tag{117}$$

Fig. 20 is the μ plot to test condition (99). As expected, the value μ approaches 1 at zero frequency. We see that $\mu << 1$ for all frequencies away from $\omega = 0$. As expected, μ rapidly increases towards 1 at very low frequencies because the integrators cause a stability problem here as either of the ϵ_i approach zero. Since $\mu \leq 1$, the system demonstrates RDUS.

RDD vs. RDUS Let us look at the set of controllers which are given by dynamically detuning the system. This set is

$$K = (\tilde{S}\tilde{P})^{-1}(I - \tilde{S}) = \frac{75s + 1}{s} \begin{bmatrix} -\frac{1}{0.878\lambda_1} & 0 \\ 0 & -\frac{1}{0.014\lambda_2} \end{bmatrix}. \tag{118}$$

We see that for this example dynamically detuning the system exactly corresponds to decreasing the single loop controller gains. Thus, for this example, RDD is equivalent to the interior μ test (99) for RDUS. This will not be true in general.

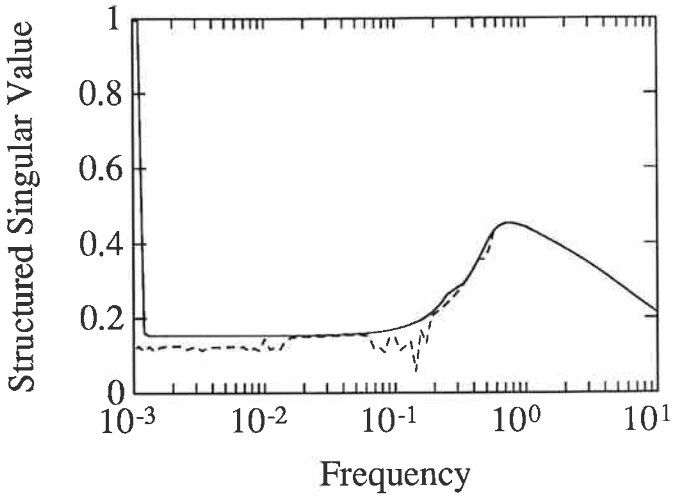

Figure 20: Structured singular value plot (upper and lower bounds) to test for RDUS in Example 3.

8 Conclusions

The complete set of robust loopshaping bounds have been derived for mixed real/complex uncertainties. Low-order robust controllers can be designed by loopshaping open loop transfer functions. Closed loop transfer functions can be loopshaped to meet gain and phase margins specifications. Robust controllers that meet multiple performance specifications are designed by using the most restrictive bounds for loopshaping. Necessary and sufficient tests were derived for versions of fault and failure tolerance that have been defined previously in the literature. It was shown how to design robust fault/failure tolerant controllers via loopshaping. The technique was applied to design a failure and fault tolerant decentralized controller for a high-purity distillation column.

9 Future Work

The robust loopshaping bounds can be calculated with off-the-shelf software whenever the controller is SISO or fully-decentralized. More work is needed to enable the calculation of the bounds in the general case. These bounds have wider applicability than just for controller design—Braatz et al [5] showed that the loopshaping bounds have direct application to the problem of robust control structure selection (i.e. choosing actuators, sensors, and pairings for control purposes).

It would be useful to have more examples which use robust loopshaping to design controllers. In particular, an example with mixed real/complex uncertainty is need. More multivariable examples are needed—especially interesting would be loopshaping decentralized PID controllers using the diagonal open loop transfer function $\tilde{L}$. Controllers designed via loopshaping are less complicated and more intuitive than designs provided by DK-iteration. On the other hand, DK-iteration may give controllers with a smaller μ value. It would be useful to compare designs from the two methods. The authors are currently working on this.

When K and G are both stable, simple sufficient tests exist for determining integrity (see Fujita and Shimemura [10]). Since CDUS implies integrity, Thm. 7.4 (with the $\mu \leq 1$ test replaced by $\mu < 1$, see Remark 7.7) gives a sufficient test for integrity when K and G are stable. It should be investigated if this test is less conservative for determining system integrity than the tests in [10]. This is currently being investigated.

10 Acknowledgements

The authors acknowledge Morten Hovd for deriving Lemma 3.7 and for discussions involving robust fault/failure tolerance. The first author thanks the American-Scandinavian Association of Los Angeles and the Norway-America Association in Oslo, Norway for support during the author's stay in Trondheim during the summer of 1992; and he thanks the Fannie and John Hertz Foundation for their continued support.

References

[1] G. J. Balas, A. K. Packard, J. C. Doyle, K. Glover, and R. S. R. Smith. Development of advanced control design software for researchers and engineers. In *Proceedings of the 1991 American Control Conference*, pages 996–1001, 1991.

[2] H. W. Bode. *Network Analysis and Feedback Amplifier Design.* D. Van Nostrand, Princeton, N.J., 1945.

[3] R. D. Braatz and M. Morari. Robust control for a noncolocated spring-mass system. *AIAA Journal of Guidance, Control and Dynamics*, 1992. in press.

[4] R. D. Braatz and M. Morari. Robust control for a noncolocated spring-mass system. In *Proceedings of the 1992 American Control Conference*, pages 2061–2062, 1992.

[5] R. D. Braatz, M. Morari, and J. H. Lee. Necessary, and sufficient loop-shaping bounds for robust performance. Presented at the *AIChE Annual Meeting*, Paper 154d, 1991.

[6] P. J. Campo and M. Morari. Achievable closed loop properties of systems under decentralized control: Conditions involving the steady state gain. 1991. submitted.

[7] J. C. Doyle. Analysis of feedback systems with structured uncertainties. *IEE Proceedings Part D*, 129:242–250, 1982.

[8] J. C. Doyle. Structured uncertainty in control system design. In *Proceedings of the 24th IEEE Conference on Decision and Control*, pages 260–265, 1985.

[9] J. C. Doyle, F. A. Francis, and A. Tannenbaum. *Feedback Control design*, chapter 6,7. MacMillan, 1991.

[10] M. Fujita and E. Shimemura. Integrity against arbitrary feedback loop failure in linear multivariable control systems. *Automatica*, 24(6):765–772, 1988.

[11] M. Hovd. Studies on control structure selection and design of robust decentralized and svd controllers, 1992. Dr. Ing. Thesis, University of Trondheim-NTH, Trondheim, Norway.

[12] P. Jackson. Applying μ-synthesis to missile autopilot design. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 2993–2998, 1990.

[13] D. K. Le, O. D. I. Nwokah, and A. E. Frazho. Multivariable decentralized integral controllability. *International Journal of Control*, 54:481–496, 1991.

[14] J. H. Lee, R. D. Braatz, M. Morari, and A. Packard. Screening tools for robust control structure selection. Presented at the *AIChE Annual Meeting*, Paper 152p, 1991.

[15] M. Morari and E. Zafiriou. *Robust Process Control.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1989.

[16] O. D. I. Nwokah and R. Perez. On multivariable stability in the gain space. In *Proceedings of the 1990 IEEE Conference on Decision and Control*, pages 328–333, 1990.

[17] O. D. I. Nwokah and R. Perez. On multivariable stability in the gain space. *Automatica*, 27:975–983, 1991.

[18] S. Skogestad and M. Morari. Some new properties of the structured singular value. *IEEE Trans. on Auto. Control*, AC-33:1151–1154, 1988.

[19] S. Skogestad and M. Morari. Robust performance of decentralized control systems by independent designs. *Automatica*, 29:119–125, 1989.

[20] S. Skogestad, M. Morari, and J. C. Doyle. Robust control of ill-conditioned plants: High purity distillation. *IEEE Trans. on Auto. Control*, AC-33:1092–1105, 1988.

[21] R. J. Veillette, J. V. Medanić, and W. R. Perkins. Design of reliable control systems. *IEEE Trans. on Auto. Control*, 37:290–304, 1992.

[22] M. Vidyasagar. *Control System Synthesis: A Factorization Approach.* MIT Press, Cambridge, Massachusetts, 1985.