Output multiplicative uncertainty

Theorem

Under assumptions (A1), (A2), (A3), the closed-loop system in figure() is stable if:

$$\forall \omega \in \mathbb{R}\ ,\ \|\Delta_s(j\omega)\| \left\| G(j\omega)K(j\omega)\left[I_p + G(j\omega)K(j\omega)\right]^{-1} \right\| < 1, \qquad (0.139)$$

or, equivalently, $\left\| G(j\omega)K(j\omega)(I_p + G(j\omega)K(j\omega))^{-1} \right\| < \delta_m^{-1}(\omega)$. *(0.140)*

Under assumption (A4), the closed-loop system in figure() is stable if and only if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\| G(j\omega)K(j\omega)(I_p + G(j\omega)K(j\omega))^{-1} \right\| < \delta_m^{-1}(\omega). \qquad (0.141)$$

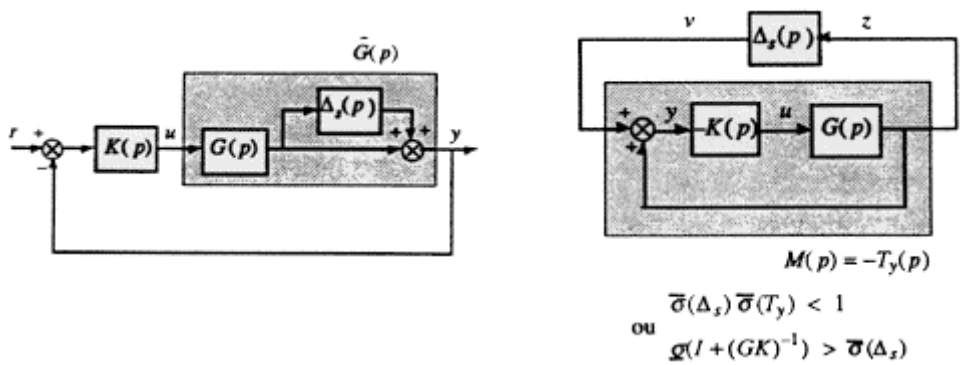


Figure 26: Equivalent $M - \Delta$ interconnection for output multiplicative uncertainty

This last condition can be written as: $\|T_y(j\omega)\| < \delta_m^{-1}(\omega)$, where

$$T_y(s) := G(s)K(s)\left[I_p + G(s)K(s)\right]^{-1} \qquad (0.142)$$

is the output complementary sensitivity matrix. We can see from the robust stability condition for a multiplicative uncertainty, that T_y reflects the capacity of the system to tolerate uncertainty of the output multiplicative form.

4.2.3 Robust stability with inverse multiplicative uncertainty

Input inverse multiplicative uncertainty

Theorem

Under assumptions (A1), (A2), (A3), the closed-loop system in figure() is stable if:

$$\forall \omega \in \mathbb{R}\ ,\ \|\Delta_s(j\omega)\| \left\| \left[I_m + K(j\omega)G(j\omega) \right]^{-1} \right\| < 1\,, \qquad (0.143)$$

or, equivalently,

$$\left\| \left[I_m + K(j\omega)G(j\omega) \right]^{-1} \right\| < \delta_m^{-1}(\omega)\,. \qquad (0.144)$$

Under assumption (A4), the closed-loop system in figure() is stable if and only if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\| \left[I_m + K(j\omega)G(j\omega) \right]^{-1} \right\| < \delta_m^{-1}(\omega)\,. \qquad (0.145)$$

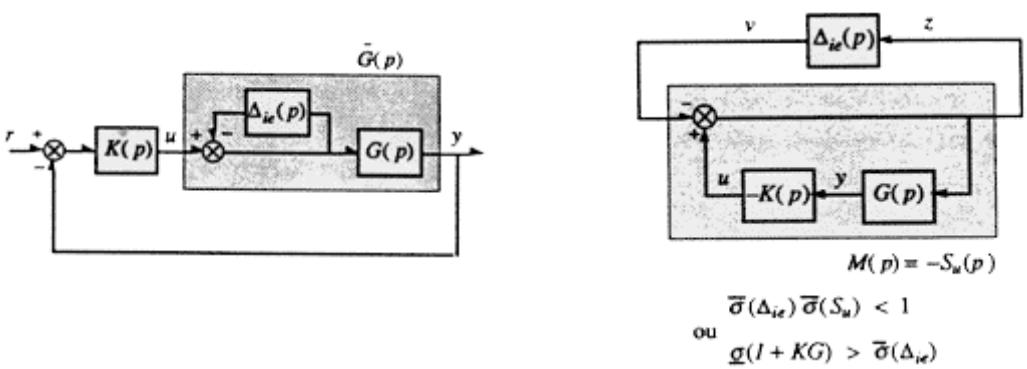


Figure 27: Equivalent $M - \Delta$ interconnection for input inverse multiplicative uncertainty

Output inverse multiplicative uncertainty

Theorem

Under assumptions (A1), (A2), (A3), the closed-loop system in figure() is stable if:

$$\forall \omega \in \mathbb{R}\ ,\ \|\Delta_s(j\omega)\| \left\| \left[I_p + G(j\omega)K(j\omega) \right]^{-1} \right\| < 1\,, \qquad (0.146)$$

or, equivalently,

$$\left\| \left[I_p + G(j\omega)K(j\omega) \right]^{-1} \right\| < \delta_m^{-1}(\omega)\,. \qquad (0.147)$$

Under assumption (A4), the closed-loop system in figure() is stable if and only if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\| \left[I_p + G(j\omega)K(j\omega) \right]^{-1} \right\| < \delta_m^{-1}(\omega)\,. \qquad (0.148)$$

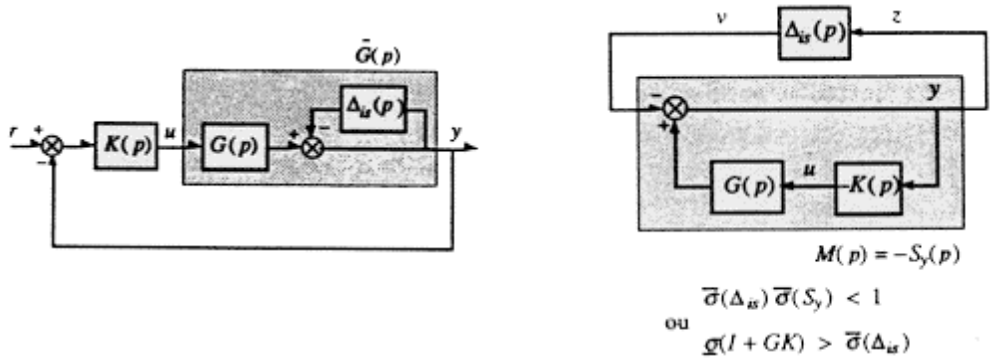


Figure 28: Equivalent $M - \Delta$ interconnection for output inverse multiplicative uncertainty

4.2.4 Robust stability with feedback uncertainty

Theorem

Under assumptions (A1), (A2), (A3), the closed-loop system in figure() is stable if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\|\Delta_f(j\omega)\right\| \left\|\left[I_p + G(j\omega)K(j\omega)\right]^{-1} G(j\omega)\right\| < 1, \qquad (0.149)$$

or, equivalently,

$$\left\|\left[I_p + G(j\omega)K(j\omega)\right]^{-1} G(j\omega)\right\| < \delta_f^{-1}(\omega). \qquad (0.150)$$

Under assumption (A4), the closed-loop system in figure() is stable if and only if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\|\left[I_p + G(j\omega)K(j\omega)\right]^{-1} G(j\omega)\right\| < \delta_f^{-1}(\omega). \qquad (0.151)$$

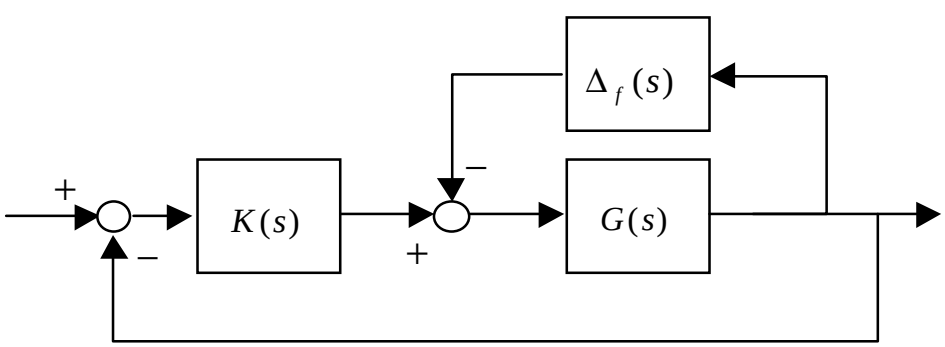


Figure 29: Interconnection for feedback uncertainty

4.2.5 Robust stability with linear fractional uncertainty

Theorem

Under assumptions (A1), (A2), (A3), the closed-loop system in figure() is stable if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\|\Delta_s(j\omega)\right\| \left\|\mathcal{F}_L\left[P(j\omega), K(j\omega)\right]\right\| < 1, \qquad (0.152)$$

or, equivalently,

$$\left\|\mathcal{F}_L\left[P(j\omega), K(j\omega)\right]\right\| < \delta_l^{-1}(\omega). \qquad (0.153)$$

Under assumption (A4), the closed-loop system in figure() is stable if and only if:

$$\forall \omega \in \mathbb{R}\ ,\ \left\|\mathcal{F}_L\left[P(j\omega), K(j\omega)\right]\right\| < \delta_l^{-1}(\omega). \qquad (0.154)$$

The figure below shows a feedback controlled upper LFT with the perturbation. This is the most general representation as all other robust stability conditions can obtained from this one. Typically, the generalized plant $P(s)$ would embed all of the weighting functions used to specify performance and also weighting functions that shape the uncertainty.

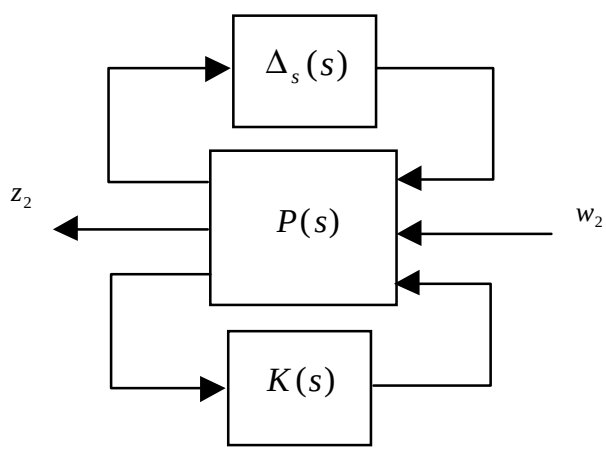


Figure 30: LFT for linear fractional uncertainty

The first condition given in each theorem above are just sufficient conditions for robust stability. But the second condition given based on the small-gain theorem are necessary and sufficient, and these are the most useful to control engineers. As previously mentioned, the fact that $\Delta(s)$ must be stable in this case is not too restrictive.

The uncertainty bound $\delta(\omega)$ is usually specified by a *scalar* stable rational weigthing function $W(s)$ as follows. Let the perturbation be defined as $\Delta(s) := W(s)\tilde{\Delta}(s)$ where $\tilde{\Delta}(s)$ is a normalized perturbation in $\mathcal{RH}_\infty$ such that $\left\|\tilde{\Delta}(s)\right\|_\infty < 1$. Then

$$\begin{gathered}
\left\|\Delta(j\omega)\right\| < \left|W(j\omega)\right| = \delta(\omega), \quad \forall \omega \in \mathbb{R} \\
\Leftrightarrow \\
\left\|W^{-1}(j\omega)\Delta(j\omega)\right\| = \left\|\tilde{\Delta}(j\omega)\right\| < 1, \quad \forall \omega \in \mathbb{R} \\
\Leftrightarrow \\
\left\|\tilde{\Delta}(s)\right\|_\infty < 1
\end{gathered} \tag{0.155}$$

Thus, the upper bound $\left\|\tilde{\Delta}(s)\right\|_\infty < 1$ on the $\mathcal{H}_\infty$-norm of the normalized perturbation can be used in the theorems, along with the weighting functions.

These conditions are listed in the table below.

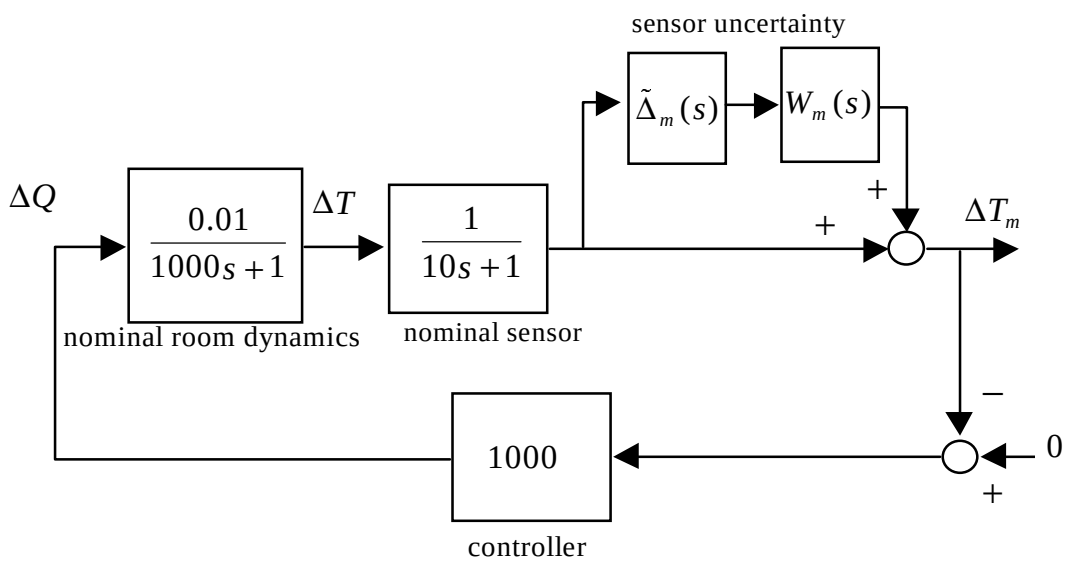
Table 1: Robust stability conditions

Uncertainty form	**Assumptions** $W_x, \tilde{\Delta}_x \in \mathcal{RH}_\infty$	**Necessary and sufficient condition for robust stability**
Additive	$\Delta_a = W_a \tilde{\Delta}_a$, $\left\|\tilde{\Delta}_a\right\|_\infty < 1$	$\left\|KS_y W_a\right\|_\infty = \left\|K(I_p + GK)^{-1} W_a\right\|_\infty \le 1$
Input multiplicative	$\Delta_m = W_m \tilde{\Delta}_m$, $\left\|\tilde{\Delta}_m\right\|_\infty < 1$	$\left\|T_u W_m\right\|_\infty = \left\|KG\left[I_m + KG\right]^{-1} W_m\right\|_\infty \le 1$
Output multiplicative	$\Delta_m = W_m \tilde{\Delta}_m$, $\left\|\tilde{\Delta}_m\right\|_\infty < 1$	$\left\|T_y W_m\right\|_\infty = \left\|GK\left[I_p + GK\right]^{-1} W_m\right\|_\infty \le 1$
Input inverse multiplicative	$\Delta_i = W_i \tilde{\Delta}_i$, $\left\|\tilde{\Delta}_i\right\|_\infty < 1$	$\left\|S_u W_i\right\|_\infty = \left\|KG\left[I_m + KG\right]^{-1} W_i\right\|_\infty \le 1$
Output inverse multiplicative	$\Delta_i = W_i \tilde{\Delta}_i$, $\left\|\tilde{\Delta}_i\right\|_\infty < 1$	$\left\|S_y W_i\right\|_\infty = \left\|\left[I_p + GK\right]^{-1} W_i\right\|_\infty \le 1$
Feedback	$\Delta_f = W_f \tilde{\Delta}_f$, $\left\|\tilde{\Delta}_f\right\|_\infty < 1$	$\left\|S_y G W_f\right\|_\infty = \left\|\left[I_p + GK\right]^{-1} G W_f\right\|_\infty \le 1$
LFT	$\Delta_l = W_l \tilde{\Delta}_l$, $\left\|\tilde{\Delta}_l\right\|_\infty < 1$ W_l absorbed in P	$\left\|\mathcal{F}_L\left[P, K\right]\right\|_\infty \le 1$

These results indicate that we have to minimize the $\mathcal{H}_\infty$-norm of certain closed-loop transfer matrices in order to increase robustness margins, i.e., the level of uncertainty that can be tolerated without destabilizing the system.

Example: Robust stability of room heating process control with uncertain sensor dynamics

The temperature sensor with first-order dynamics was assumed to have 10% uncertainty in its frequency response, i.e., $\|\Delta_I(j\omega)\| < 0.1,\ \forall\omega$, so that $W_m(s) = 0.1$. This uncertainty was modeled with an output multiplicative model, which was then transformed into an LFT model. The controller $K(s) = 1000$ is chosen as a pure gain.



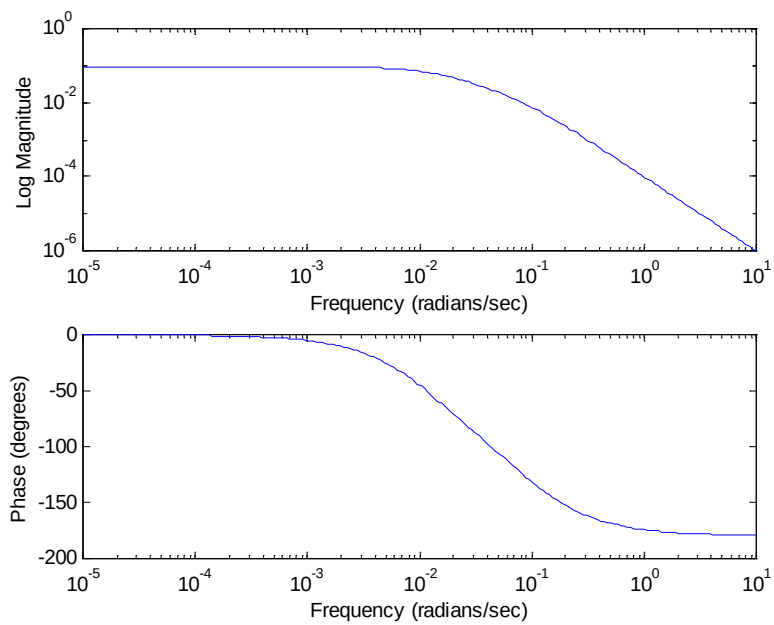
Note that the controller stabilizes the nominal loop gain. The condition for robust stability with output multiplicative uncertainty is $\|T_y W_m\|_\infty = \left\|GK\left[I_p + GK\right]^{-1} W_m\right\|_\infty \le 1$. Let's compute the norm of this closed-loop transfer function (from the output of the weighted uncertainty to its input):

$$
\begin{aligned}
-GK\left[I_p + GK\right]^{-1} W_m &= \frac{-10(0.1)}{(1000s+1)(10s+1)}\left(1 + \frac{10}{(1000s+1)(10s+1)}\right)^{-1} \\
&= \frac{-1}{(1000s+1)(10s+1)+10} = \frac{-1}{10000s^2 + 1010s + 11} \qquad (0.156) \\
&= \frac{-0.09}{(80.5s+1)(11.3s+1)}
\end{aligned}
$$

The absolute value of the DC gain of this transfer function is 0.09, which is also the maximum magnitude of its frequency response. Therefore,

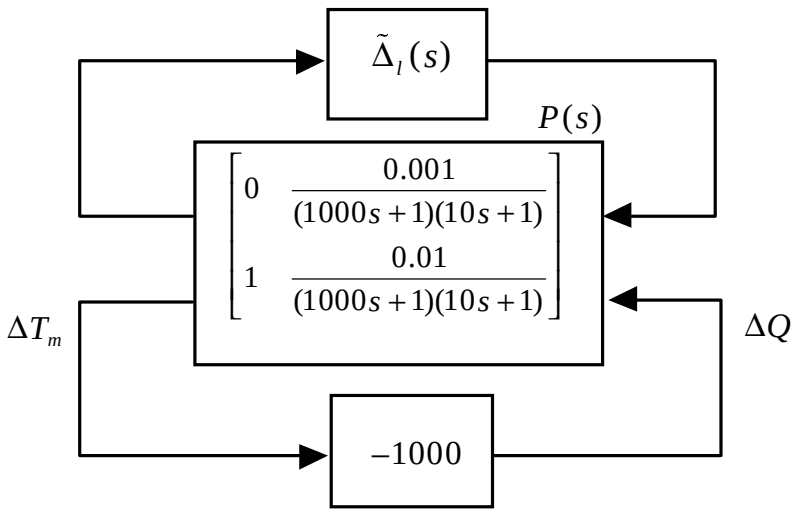
$$\left\| GK\left[I_p + GK\right]^{-1} W_m \right\|_\infty = \left\| \frac{0.09}{(80.5s+1)(11.3s+1)} \right\|_\infty = \sup_{\omega \in \mathbb{R}} \left| \frac{0.09}{(80.5j\omega+1)(11.3j\omega+1)} \right| = 0.09 < 1 \tag{0.157}$$

and hence the closed-loop system has robust stability. It also means that the closed-loop system could tolerate up to 11 (=1/0.9) times more uncertainty in the sensor before the possible onset of instability. The Bode plot of $GK\left[I_p + GK\right]^{-1} W_m$ is shown below.



We will now check that the robustness condition based on the LFT representation will give us the same conclusion. This condition is $\left\| \mathcal{F}_L\left[P, K\right] \right\|_\infty \le 1$, where the generalized plant $P(s)$ was obtained in a previous example (but $W_l(s)$ must be added to it):

$$P_{22}(s) = G(s), \quad P_{21}(s) = I, \quad P_{11}(s) = 0, \quad P_{12}(s) = G(s)W_l(s) \tag{0.158}$$



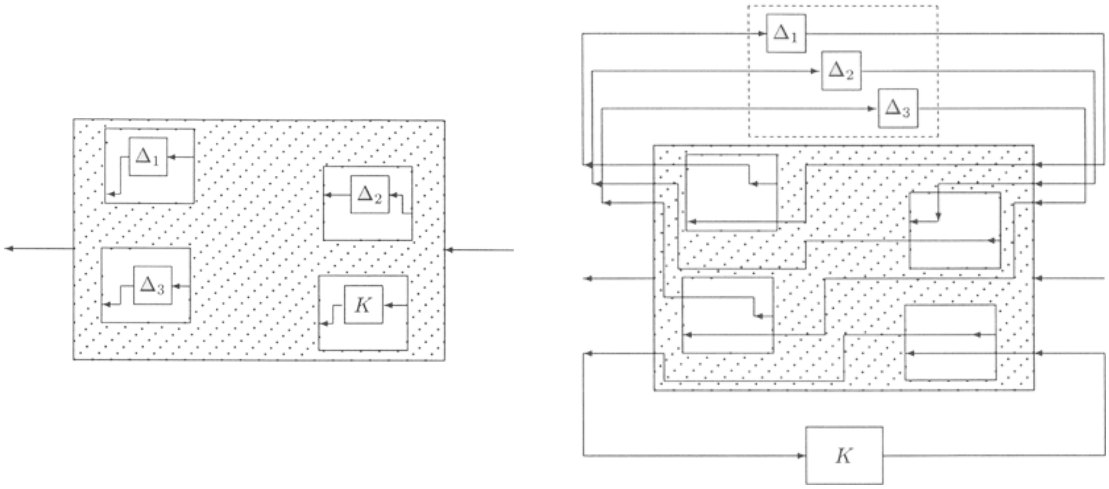
$$\begin{aligned}\mathcal{F}_L\left[P(s),K(s)\right] &= P_{11}(s)+P_{12}(s)K(s)\left[I-P_{22}(s)K(s)\right]^{-1}P_{21}(s)\\ &= \frac{-1}{(1000s+1)(10s+1)}\left[1+\frac{10}{(1000s+1)(10s+1)}\right]^{-1}\\ &= \frac{-1}{(1000s+1)(10s+1)+10} = \frac{-1}{10000s^2+1010s+11}\end{aligned} \tag{0.159}$$

As we expected, we get the same closed-loop transfer function, and therefore we get robust stability since

$$\left\|\mathcal{F}_L\left[P,K\right]\right\|_\infty = 0.09 < 1. \tag{0.160}$$

4.3 Structured uncertainty

Sometimes the uncertainty in a model will come from different sources: uncertain time delay, neglected hig-frequency dynamics, actuator uncertainty, etc. One way to model this uncertainty is to try to "cover" all of these possible variations with a single full complex block at each frequency (i.e., an unstructured uncertainty block.) This approach often turns out to be conservative in the sense that the chosen unstructured uncertainty will have a large size and hence may generate a much larger set of perturbed plant models than what is required.



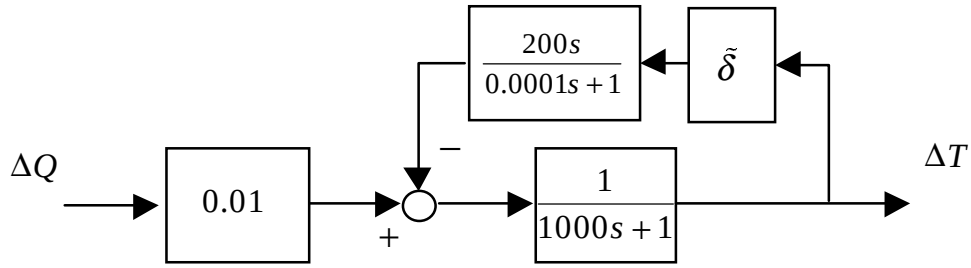
Another approach is to model each source of uncertainty separately with its own perturbation. To each perturbation will correspond a delta block in the block diagram of the perturbed plant model. The best framework to use in this case is the LFT form with a so-called *structured perturbation.* We proceed with an example to show how such a structured perturbation can be constructed.

Example: Structured uncertainty for the room heating process

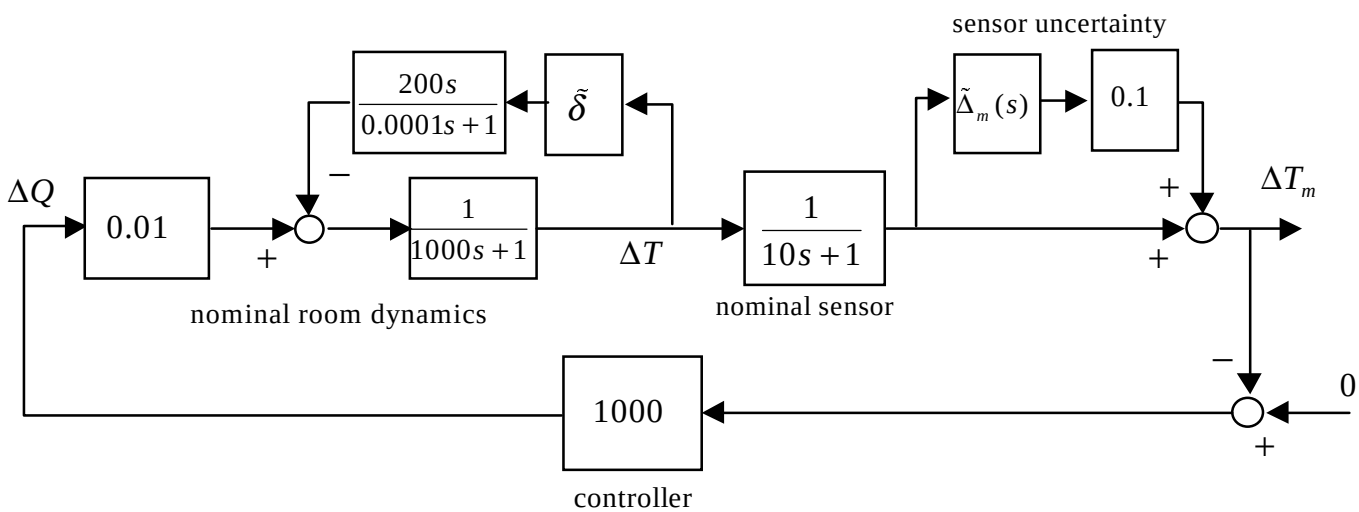
Suppose that there is uncertainty not only in the sensor, but also in the process model itself, particularly in the time constant. Assume that there is 20% uncertainty in this time constant. The additional perturbation can be "pulled out" as follows:

$$\frac{0.01}{(1000+200\tilde{\delta})s+1} = 0.01\frac{\dfrac{1}{1000s+1}}{1+\dfrac{200\tilde{\delta}s}{1000s+1}}, \quad \tilde{\delta}\in\mathbb{R}, \left|\tilde{\delta}\right|<1. \tag{0.161}$$

This equation is just a feedback interconnection with the perturbation in the feedback path, as shown below:



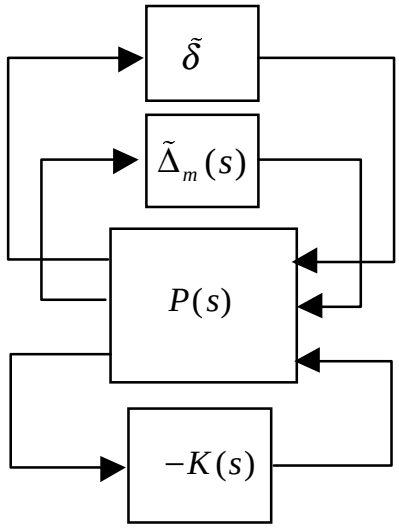
Notice that the weighting function was taken to be $\frac{200s}{0.0001s+1}$ instead of $200s$ as (0.161) suggests. This is because because all weighting functions must be in $\mathcal{RH}_\infty$, and $200s$ isn't. The addition of the high-frequency pole should not affect the design at frequencies of interest. Plugging the above diagram into the overall perturbed plant block diagram, we get:



Define the structured perturbation

$$\Delta_s(s) := \begin{bmatrix} \tilde{\delta} & 0 \\ 0 & \tilde{\Delta}_m(s) \end{bmatrix}. \tag{0.162}$$

The block diagram above can be rearranged into an LFT form. A good technique to find the block entries of the generalized plant transfer matrix is to follow the signal paths on the block diagram from and to the perturbations, and from and to the controller.



We have

$$P(s)=\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix}$$

$$P_{11}(s)=\frac{-200s}{(1000s+1)(0.0001s+1)}$$

$$P_{12}(s)=0$$

$$P_{13}(s)=\frac{0.01}{1000s+1}$$

$$P_{21}(s)=\frac{-200s}{(1000s+1)(10s+1)(0.0001s+1)}$$

$$P_{22}(s)=0$$

$$P_{23}(s)=\frac{0.01}{(1000s+1)(10s+1)}$$

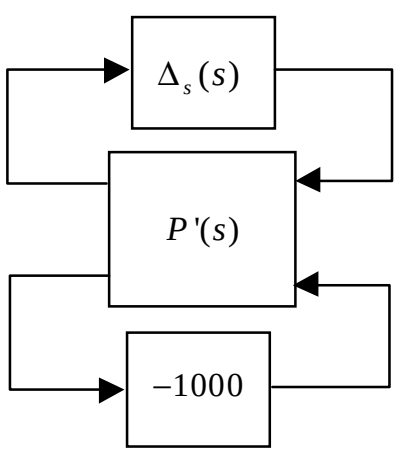
$$P_{31}(s)=\frac{-200s}{(1000s+1)(10s+1)(0.0001s+1)}$$

$$P_{32}(s)=0.1$$

$$P_{33}(s)=\frac{0.01}{(1000s+1)(10s+1)}$$

With the structured perturbation defined above, the LFT can be rearranged into another simpler LFT as shown below, where

$$P'(s)=\begin{bmatrix} P'_{11} & P'_{12} \\ P'_{21} & P'_{22} \end{bmatrix},\quad P'_{11}=\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix},\quad P'_{12}=\begin{bmatrix} P_{13} \\ P_{23} \end{bmatrix},\quad P'_{21}=\begin{bmatrix} P_{31} & P_{32} \end{bmatrix},\quad P'_{22}=P_{33}$$



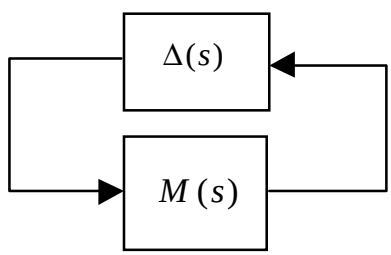
4.4 Robust closed-loop stability with structured uncertainty: μ -Analysis

Robust stability analysis with structured uncertainty was developed by Doyle (Doyle, 1982). He introduced the *structured singular value* of a matrix as a tool of linear algebra that can be used to analyze stability robustness with structured uncertainty in the frequency domain. This seminal work has developed into a theory called μ -analysis. The robust control design counterpart of this theory is called μ -synthesis and will be studied in the next chapter.

4.4.1 The structured singular value

Basic concept

The structured singular value is a generalization of the maximum singular value of constant complex matrices. Let us consider again the robust stability problem of the following standard feedback interconnection with stable $M(s)$ and $\Delta(s)$.



The closed-loop poles are given by $\det\left[I - M(s)\Delta(s)\right] = 0$, and the feedback system becomes unstable if $\det\left[I - M(s)\Delta(s)\right] = 0$ for some $s \in \overline{\mathbb{C}}_+$ (closed RHP). Let $\beta > 0$ a sufficiently small number such that the closed-loop system is stable for all stable perturbation bounded by $\|\Delta\|_\infty < \beta$. Next, increase β up to $\beta_{\max}$ such that the closed-loop system becomes unstable for some $\|\Delta\|_\infty < \beta_{\max}$. So $\beta_{\max}$ is the robust stability margin. By the small-gain theorem,

$$\frac{1}{\beta_{\max}} = \|M\|_\infty := \sup_{s \in \overline{\mathbb{C}}_+} \bar{\sigma}\left(M(s)\right) = \sup_{\omega} \bar{\sigma}\left(M(j\omega)\right) \tag{0.163}$$

if $\Delta(s)$ is unstructured.

In fact, we can tell how large the bound on $\Delta(s)$ can get in terms of the $\mathcal{H}_\infty$ -norm of $M(s)$ such that stability of the closed-loop system will be maintained. Namely, the feedback system can tolerate uncertainty with $\mathcal{H}_\infty$ -norm up to $\|M\|_\infty^{-1}$.

Note that for any fixed complex number $s_0 \in \overline{\mathbb{C}}_+$, the maximum singular value (the norm) $\bar{\sigma}\left(M(s)\right)$ can be written as:

$$\left\|M\left(s_0\right)\right\| = \bar{\sigma}\left(M\left(s_0\right)\right) = \frac{1}{\min\left\{\bar{\sigma}\left(\Delta\right) : \det\left(I - M\left(s_0\right)\Delta\right) = 0, \Delta \in \mathbb{C}^{p\times p}\right\}} \tag{0.164}$$

The reciprocal of the *largest singular value* of $M\left(s_0\right)$ is a measure of the smallest unstructured $\Delta \in \mathbb{C}^{p\times p}$ that can cause instability of the feedback system.

To quantify the smallest destabilizing *structured* complex Δ_s, the concept of maximum singular value needs to be generalized. We loosely define (we'll give a more formal definition later)

$$\mu_\Delta\left(M\left(s_0\right)\right) = \frac{1}{\min\left\{\bar{\sigma}\left(\Delta_s\right) : \det\left(I - M\left(s_0\right)\Delta_s\right) = 0,\ \Delta_s \text{ is structured}\right\}}$$

as the largest structured singular value of $M\left(s_0\right)$ with respect to the structured complex perturbation Δ_s. Then, the robust stability margin $\beta_{\max}$ of the feedback system with structured complex uncertainty Δ_s is given by: $\dfrac{1}{\beta_{\max}} = \sup\limits_{s\in\overline{\mathbb{C}}_+} \mu_\Delta\left(M\left(s\right)\right) = \sup\limits_{\omega\in\mathbb{R}} \mu_\Delta\left(M\left(j\omega\right)\right)$.

Definition of the structured singular value μ

We consider complex matrices $M \in \mathbb{C}^{nxn}$. In the definition of $\mu\left(M\right)$, there is an underlying structure $\mathbf{\Delta}$ (set of structured, block-diagonal matrices) on which everything in the sequel depends. For each problem, this structure is, in general, different; it depends on the *uncertainty* and *performance objectives* of the problem. Defining the structure involves specifying three things: the type of each block, the total number of blocks, and their dimensions.

There are two types of blocks: (*possibly-repeated*) *scalar* blocks and *full* blocks. Two nonnegative integers, S and F, represent the number of (*possibly-repeated*) *scalar* blocks and the number of *full* blocks, respectively.

We consider the general case of a S-scalar-block, F-full-block structured perturbation, where the scalar blocks may be repeated. Let $M \in \mathbb{C}^{n\times n}$, and $r_1,\ldots,r_q, m_1,\ldots,m_l$ be positive integers satisfying $\sum_{i=1}^{S} r_i + \sum_{j=1}^{F} m_j = n$ (this is just for bookkeeping of the dimensions of the uncertainty blocks).

We define the block structure $\Gamma \subset \mathbb{C}^{n\times n}$ as

$$\Gamma := \left\{\Delta_s = \operatorname{diag}\left\{\delta_1 I_{r_1},\ldots,\delta_q I_{r_q},\Delta_1,\ldots,\Delta_l\right\} : \delta_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j\times m_j}\right\}. \tag{0.165}$$

Definition: structured singular value

For a complex matrix $M \in \mathbb{C}^{n\times n}$, the function $\mu_\Gamma : \mathbb{C}^{n\times n} \to \mathbb{R}_+$ is defined as

$$\mu_\Gamma(M) := \left[\min\left\{\|\Delta_s\| : \Delta_s \in \Gamma,\ \det(I - M\Delta_s)\right\} = 0\right]^{-1} \tag{0.166}$$

unless no $\Delta_s \in \Gamma$ makes $\det(I - M\Delta_s)$ singular, in which case $\mu_\Gamma(M) = 0$.

Remarks

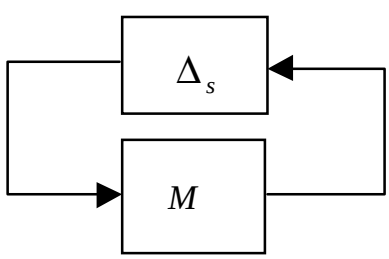
- This definition is for complex matrices, *not for transfer matrices*. $\mu_\Gamma(M)$ is called the structured singular value because, as previously explained, if the structure is simplified to a single-block complex uncertainty, then $\mu_\Gamma(M) = \bar{\sigma}(M)$, the maximum singular value of the matrix. Hence, $\mu_\Gamma(M)$ is a generalization of the maximum singular value (or spectral norm) of matrix M.

- In words, $\mu_\Gamma(M)$ is the inverse of the size of the smallest structured perturbation that makes $I - M\Delta_s$ singular. Thus, this definition is directly related to the small-gain theorem.

- The full uncertainty blocks do not have to be square. We assume that they are for ease of presentation.

- Computation of the structured singular value is usually performed frequency-by-frequency on the frequency response of a system.

Suppose $M \in \mathbb{C}^{n\times n}$ and consider the loop shown in the figure below.



Here, $\mu_\Gamma(M)$ is the reciprocal of the norm of the smallest structured perturbation, $\Delta_s \in \Gamma$, that causes "instability" of the feedback loop above, in the sense of singularity of the matrix $I - M\Delta_s$, or well-posedness of the corresponding closed-loop equations.

Define the " unit open ball" of complex structured perturbations:

$$\mathcal{B}\Gamma := \left\{\Delta \in \Gamma : \|\Delta\| < 1\right\}. \tag{0.167}$$

Then, an alternative expression for $\mu_\Gamma(M)$ follows from the definition:

$$\mu_\Gamma(M) = \max_{\Delta_s \in \beta\Gamma} \rho(M\Delta_s)$$

Since the spectral radius is a continuous function, the structured singular value $\mu_\Gamma : \mathbb{C}^{nxn} \to \mathbb{R}_+$ is also a continuous function. However, $\mu_\Gamma(\cdot)$ is not a norm, since it doesn't satisfy the triangle inequality.

In the special cases of only a single scalar block, and only a single full block, $\mu_\Gamma(M)$ reduces to the following.

If $\Gamma = \{\delta I : \delta \in \mathbb{C}\}$ $(S = 1, F = 0, r_1 = n)$, then

$$\mu_\Gamma(M) = \rho(M), \tag{0.168}$$

which is the spectral radius of *M*.

If $\Gamma = \mathbb{C}^{nxn}$ $(S = 0, F = 1, m_1 = n)$, then

$$\mu_\Gamma(M) = \bar{\sigma}(M). \tag{0.169}$$

Obviously, for a general uncertainty structure Γ as defined in the beginning, we must have

$$\{\delta I_n : \delta \in \mathbb{C}\} \subset \Gamma \subset \mathbb{C}^{nxn}. \tag{0.170}$$

The bigger the uncertainty set, the more possible it is for a small perturbation to make $I - M\Delta_s$ singular. We conclude that

$$\rho(M) \le \mu_\Gamma(M) \le \bar{\sigma}(M). \tag{0.171}$$

The gap between $\rho(M)$ and $\bar{\sigma}(M)$ can be large, which makes the evaluation of the structured singular value using these bounds difficult.

However, the bounds can be refined by considering transformations on M that *do not affect* $\mu_\Delta(M)$, but *do affect* $\rho(M)$ and $\bar{\sigma}(M)$. To do this, define the following two subsets of $\mathbb{C}^{nxn}$:

$$\mathcal{U} := \{U \in \Gamma : UU^* = I_n\}, \tag{0.172}$$

which is the set of structured unitary matrices, and

$$\mathcal{D} := \left\{ \begin{array}{c} \operatorname{diag}\left\{D_1, \ldots, D_s, d_1 I_{m_1}, \ldots, d_{F-1} I_{m_{F-1}}, I_{m_F}\right\}: \\ D_i \in \mathbb{C}^{r_i x r_i}, D_i = D_i^* > 0, d_j \in \mathbb{R}, d_j > 0 \end{array} \right\}. \tag{0.173}$$

which is the set of matrices that commute with any $\Delta_s \in \Gamma$. Mathematically, for any $\Delta_s \in \Gamma, U \in \mathcal{U}, D \in \mathcal{D}$, we have the following properties:

$$U^* \in \mathcal{U}, \quad U\Delta_s \in \Gamma, \quad \Delta_s U \in \Gamma, \quad \bar{\sigma}\left(U\Delta_s\right) = \bar{\sigma}\left(\Delta_s U\right) = \bar{\sigma}\left(\Delta_s\right), \tag{0.174}$$

$$D\Delta_s = \Delta_s D. \tag{0.175}$$

Consequently, we have:

For all $U \in \mathcal{U}, D \in \mathcal{D}$:

$$\mu_\Gamma\left(MU\right) = \mu_\Gamma\left(UM\right) = \mu_\Gamma\left(M\right) = \mu_\Gamma\left(DMD^{-1}\right) \tag{0.176}$$

Therefore, the lower and upper bounds on μ in (0.171) can be tightened to:

$$\max_{U \in \mathcal{U}} \rho(UM) \le \max_{\Delta_s \in \mathcal{B}\Gamma} \rho\left(\Delta_s M\right) = \mu_\Gamma\left(M\right) \le \inf_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right) \tag{0.177}$$

Bounds

The lower bound of the expression obtained above is always an equality:

$$\max_{U \in \mathcal{U}} \rho(UM) = \mu_\Gamma\left(M\right) \le \inf_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right) \tag{0.178}$$

Unfortunately, the function $\rho\left(UM\right)$ can have multiple local maxima which are not global. Thus local search cannot be guaranteed to obtain $\mu_\Gamma\left(M\right)$, but can only yield a lower bound. The upper bound can be reformulated as a convex optimization problem, so the global minimum can, in principle, be found. Unfortunately, the upper bound is not always equal to $\mu_\Gamma\left(M\right)$. For block structures Γ with a number of scalar blocks and full blocks satisfying $2S + F \le 3$, the upper bound is always equal to $\mu_\Gamma\left(M\right)$, and for block structures with $2S + F > 3$, there exist matrices for which $\mu_\Gamma\left(M\right)$ is less than the infimum. The following table gives the cases for which we have equality between the upper bound $\inf_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right)$ and the value of $\mu_\Gamma\left(M\right)$.

$$\mu_\Gamma\left(M\right) = \inf_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right) \text{if } 2S + F \le 3$$

Table 2: Number of full and scalar blocks for which the upper bound equals mu

F= / S=	0	1	2	3	4
0		Yes	yes	Yes	no
1	Yes	Yes	no	No	no
2	No	No	no	No	no

Where especially, if:

S=0, F=1 : $\mu_\Gamma(M) = \bar{\sigma}(M)$.

S=1, F=0 : $\mu_\Gamma(M) = \rho(M) = \inf_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right)$.

<u>Recap</u>

- The reciprocal of the *largest singular value* of matrix M is a measure of the smallest unstructured $\Delta \in \mathbb{C}^{p \times p}$ that can cause non-invertibility ("instability") of an $M - \Delta$ feedback interconnection.
- The *structured singular value* $\mu_\Gamma(M)$ is the inverse of the size of the smallest *structured* perturbation that can cause non-invertibility ("instability") of an $M - \Delta_s$ feedback interconnection (i.e., makes $I - M\Delta_s$ singular.)
- The structured singular value $\mu_\Gamma(M)$ can't be computed directly in general, but the lower and upper bounds $\max_{U \in \mathcal{U}} \rho(UM) \le \mu_\Gamma(M) \le \inf_{D \in \mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right)$ are usually very close, if not equal.

4.4.2 Well-posedness and the main loop theorem

We now present a technical result that is used in the proof of the robust performance theorem.

Let M be a complex matrix partitioned as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{0.179}$$

and suppose there are two defined block structures, Γ_1 and Γ_2, which are compatible in size with M_{11} and M_{22}, respectively. Define a third structure Γ as

$$\Gamma := \left\{ \begin{bmatrix} \Delta_1 & 0 \\ 0 & \Delta_2 \end{bmatrix} : \Delta_1 \in \Gamma_1, \Delta_2 \in \Gamma_2 \right\}. \tag{0.180}$$

We may compute μ with respect to three structures as follows: $\mu_1(.)$ is with respect to Γ_1, $\mu_2(.)$ is with respect to Γ_2, and $\mu_\Gamma(.)$ is with respect to Γ. By definition, the notations, $\mu_1(M_{11})$, $\mu_2(M_{22})$ and $\mu_\Gamma(M)$ all make sense.

Definition: Well-posedness of LFT

Let $\Delta_2 \in \Gamma_2$, the LFT $\mathcal{F}_L(M, \Delta_2)$ is *well-posed* if $I - M_{22}\Delta_2$ is invertible.

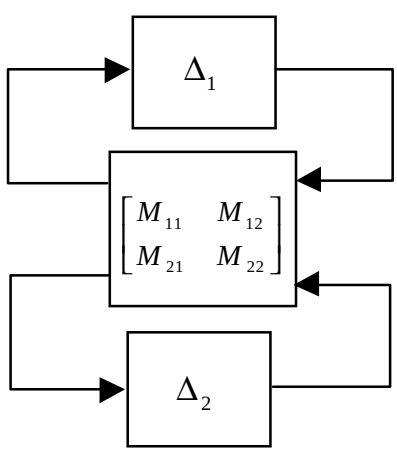
Theorem:

The linear fractional transformation $\mathcal{F}_L(M, \Delta_2)$ is well-posed for all $\Delta_2 \in \mathcal{B}\Gamma_2$ if and only if $\mu_2(M_{22}) \le 1$.

Main Loop Theorem:

The following are equivalent:

$$\mu_\Gamma(M) \le 1 \quad \Leftrightarrow \quad \begin{cases} \mu_2(M_{22}) \le 1, \text{ and} \\ \sup\limits_{\Delta_2 \in \mathcal{B}\Gamma_2} \mu_1(\mathcal{F}_L(M, \Delta_2)) \le 1. \end{cases} \tag{0.181}$$



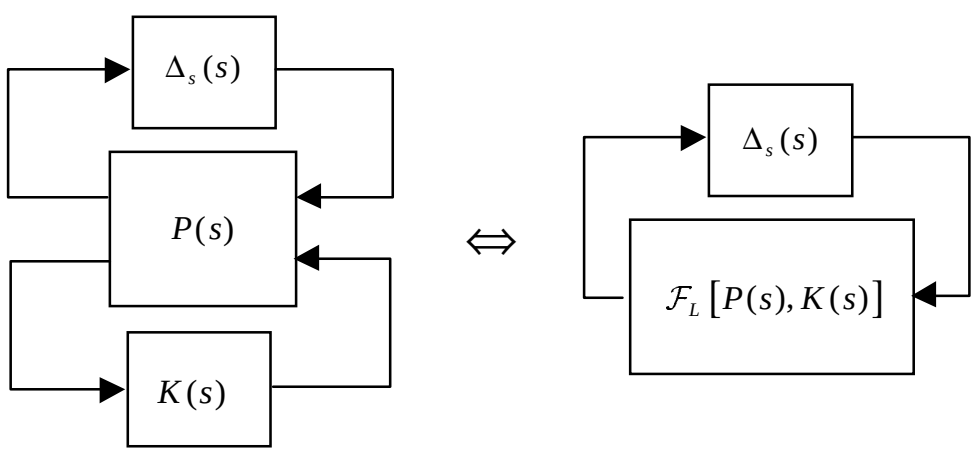
4.4.3 Robust stability with structured uncertainty

Note that so far we have defined the structured singular value and gave a number of results for complex matrices. Now, we come back to transfer matrices to give a theorem for robust closed-loop stability with structured uncertainty. We use the general framework of LFT's.

Suppose $P(s)=\begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix}$ is a generalized plant with a structured uncertainty and a controller connected to it. Let Γ be an uncertainty structure, as defined in (0.165), and assume that the dimensions of the perturbations in Γ are compatible with $P_{11}(s)$. Now define the set of real-rational, proper, stable structured perturbations:

$$\mathcal{S} := \{\Delta_s(s) \in \mathcal{RH}_\infty : \Delta_s(s_0) \in \Gamma, \forall s_0 \in \text{closed RHP}\} \tag{0.182}$$

and consider the perturbed feedback loop in the figure below where $\Delta_s(s) \in \mathcal{S}$. The LFT is simply reduced to an $M(s)-\Delta_s(s)$ interconnection and we can use a generalization of the small gain theorem to assess robust stability.



We have the following result, which is the so-called small-μ theorem:

Theorem: Robust stability with structured perturbation (small-μ theorem)

Assume controller $K(s)$ is stabilizing for the nominal plant $P(s)$. Then given $\beta > 0$, the closed-loop system in the above figure is well-posed and internally stable for all $\Delta_s \in \mathcal{S}, \|\Delta_s\|_\infty < \beta$ if and only if

$$\sup_{\omega\in\mathbb{R}} \mu_\Gamma \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\} \le \frac{1}{\beta}. \tag{0.183}$$

In particular, if the structured perturbation is normalized, i.e., $\tilde{\Delta}_s \in \mathcal{S}, \left\| \tilde{\Delta}_s \right\|_\infty < 1$, the condition becomes:

$$\sup_{\omega\in\mathbb{R}} \mu_\Gamma \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\} \le 1. \tag{0.184}$$

Remark:

- How do we check Condition (0.183) in practice? Well, we select a wide grid of frequency points covering the frequencies of interest, and we plot $\mu_\Gamma \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\}$ across these frequencies. If this plot is below β^{-1} at all frequencies, then the closed-loop system with structured uncertainty is robustly stable. If at least one point on this plot exceeds β^{-1}, then closed-loop system with structured uncertainty is *not* robustly stable.

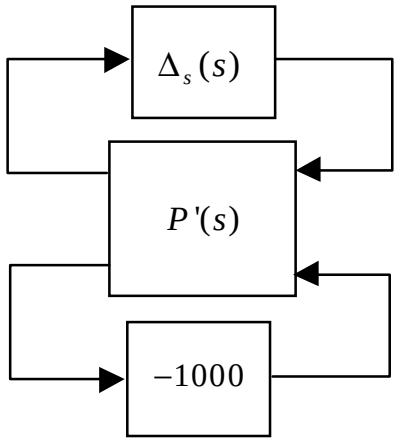
Example: Robust stability with structured uncertainty for the room heating process

Recall that in this example, the sensor dynamics and the plant's time constant were uncertain. Now the real parametric uncertainty in the time constant $\tilde{\delta} \in \mathbb{R}, \left| \tilde{\delta} \right| < 1$ can be covered with a *complex* parameter $\tilde{\delta} \in \mathbb{C}, \left| \tilde{\delta} \right| < 1$ at each frequency. Therefore we will model this uncertainty as $\tilde{\delta}(s) \in \mathcal{RH}_\infty, \left\| \tilde{\delta} \right\|_\infty < 1$.

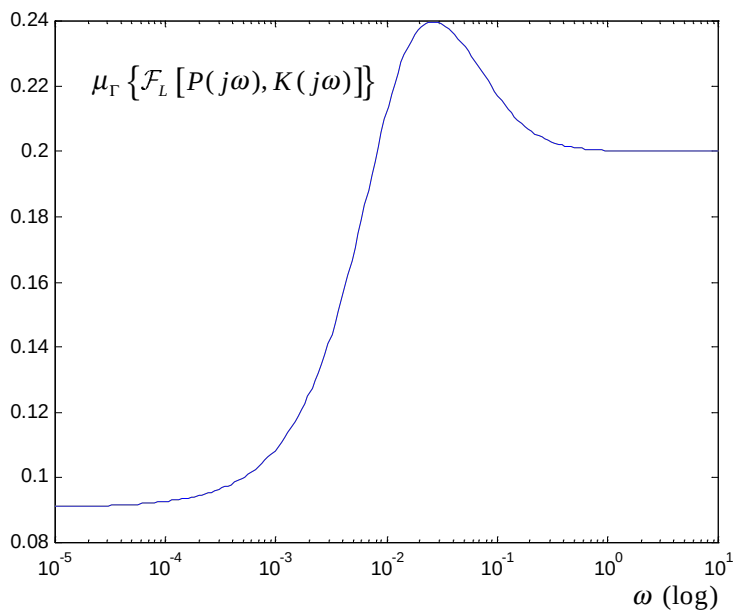
First we have to define a structured uncertainty set:

$$\Gamma := \left\{ \begin{bmatrix} \delta & 0 \\ 0 & \Delta_m \end{bmatrix} : \delta \in \mathbb{C}, \Delta_m \in \mathbb{C} \right\}. \tag{0.185}$$

Then we compute the LFT $\mathcal{F}_L \left[P'(s), K(s) \right]$ with the generalized plant and controller already given. This is tedious to do by hand, but the "sysic" and "starp" functions in Matlab facilitate the calculations, see the m-file *muroomheating.m.*

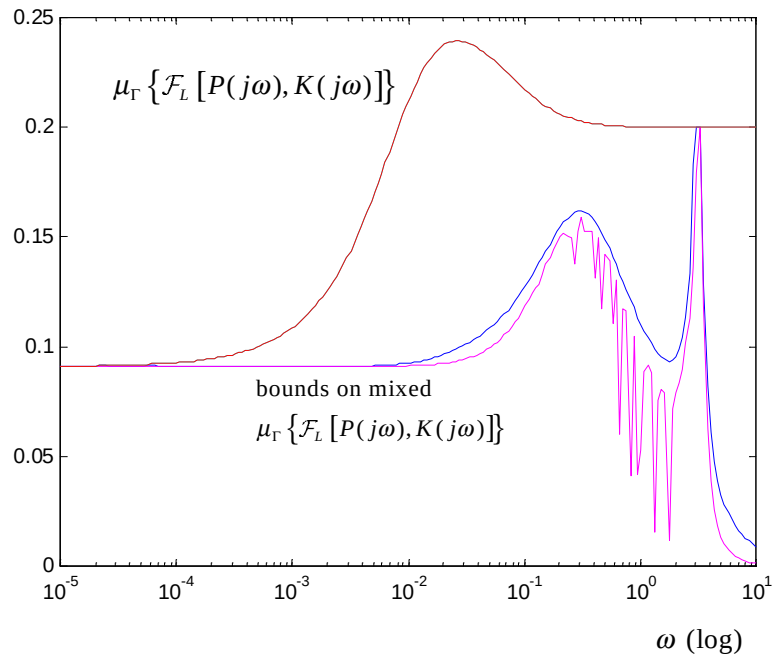


After the LFT is built, its frequency response is computed and $\mu_\Gamma\left\{\mathcal{F}_L\left[P(j\omega),K(j\omega)\right]\right\}$ is computed, or more specifically, its upper and lower bounds are computed. Even though we have two scalar blocks, these bounds are equal to each other and hence to $\mu_\Gamma\left\{\mathcal{F}_L\left[P(j\omega),K(j\omega)\right]\right\}$ in this case. The "mu-plot" is shown below, and it can be seen that $\mu_\Gamma\left\{\mathcal{F}_L\left[P(j\omega),K(j\omega)\right]\right\}$ is well below 1. Therefore the closed-loop system is robustly stable to both sensor uncertainty and uncertainty in the process time constant.



Note that mu-theory was recently enhance as the so-called "mixed-mu" theory. In this theory, *real* and complex parametric uncertainty (not just complex) can be analysed at the same time. This means that

we can leave the parametric uncertainty in the time constant as a real perturbation $\tilde{\delta} \in \mathbb{R}, \left|\tilde{\delta}\right| < 1$, instead of covering it with a *complex* parameter $\tilde{\delta} \in \mathbb{C}, \left|\tilde{\delta}\right| < 1$ at each frequency, which is more conservative. The "mu" command in Matlab can also compute bounds on mixed-μ_Γ, it suffices to specify that the first perturbation is real in the argument of the command (see *muroomheating.m*). The mixed-μ_Γ plot is shown below. As we expected, the mixed-μ_Γ bounds lie below the mu-plot (also shown on this plot) as the uncertainty set is more restricted, and hence the closed-loop system is harder to destabilize. It is of course robustly stable since the peak is below 1.



4.4.4 Robust performance with structured uncertainty

Often, stability is not the only property of a closed-loop system that must be robust to perturbations. Typically, there are exogenous disturbances acting on the system (wind gusts, sensor noise) which result in tracking and regulating errors. Under dynamic perturbations, the effect that these disturbances have on error signals can greatly increase, and therefore closed-loop performance can degrade.

Recall that nominal performance specifications can be written in terms of the $\mathcal{H}_\infty$-norms of selected closed-loop transfer matrices. Usually the system is rearranged in LFT form so that *nominal* performance is specified by a bound on the $\mathcal{H}_\infty$-norms of the LFT shown below.

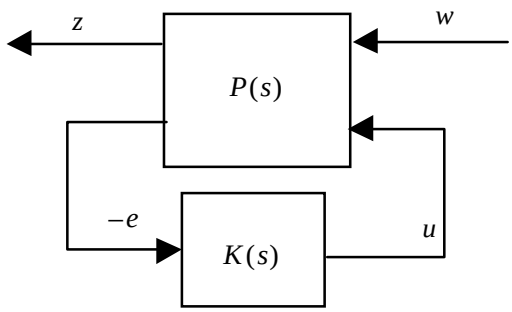


Figure 31: Standard LFT diagram for $\mathcal{H}_\infty$ -optimal control design

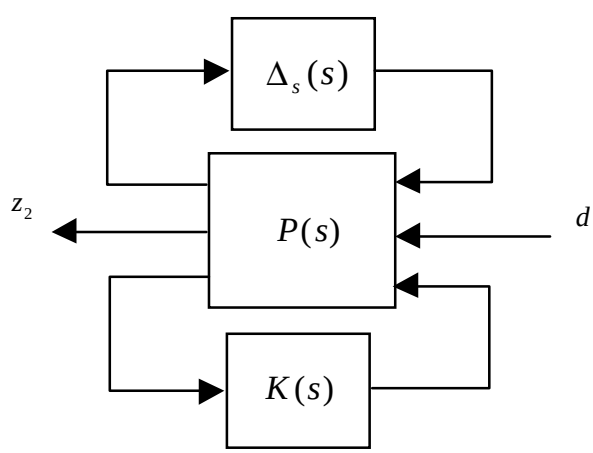
That is, with given weighting function embedded in the generalized plant $P(s)$, the performance spec is written as follows:

$$\|T_{zw}\|_\infty < 1, \tag{0.186}$$

where $T_{zw} = \mathcal{F}_L\left[P(s), K(s)\right]$.

In reality there is always some uncertainty in the model, and thus a more approriate block diagram would be the one shown below with structured uncertainty. Then the robust performance specification can be stated as follows:

$$\begin{aligned} &\left\|\mathcal{F}_L\left\{\mathcal{F}_U\left[P(s), \Delta_s(s)\right], K(s)\right\}\right\|_\infty < 1, \\ &\forall \Delta_s(s) \in \mathcal{S},\ \ \|\Delta_s(s)\|_\infty < 1 \end{aligned} \tag{0.187}$$

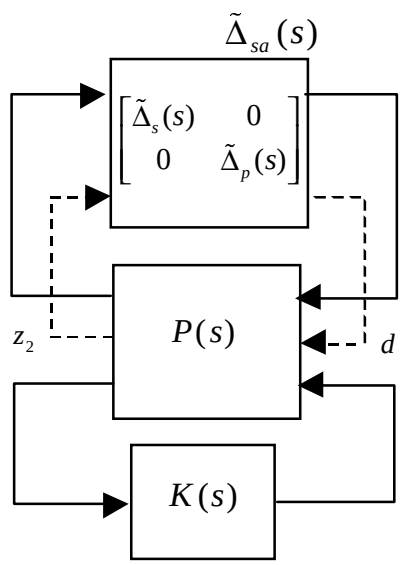


We define the robust performance criterion to be met in our design as guarantee of closed loop performance under to the perturbation acting on the system. In fact it indicates the worst-case level of performance associated with a given unstructured or structured uncertainty set.

The following result provides a means to check for robust *performance*, rather than just robust stability. Consider the perturbed closed-loop system in the figure below, where $z_2(t) \in \mathbb{R}^{n_{z2}}$ and $d(t) \in \mathbb{R}^{n_d}$, and define the augmented block structure

$$\Omega := \left\{ \Delta_{sa} = \begin{bmatrix} \Delta_s & 0 \\ 0 & \Delta_p \end{bmatrix} : \Delta_s \in \Gamma,\ \Delta_p \in \mathbb{C}^{n_d \times n_{z2}} \right\} \tag{0.188}$$

The uncertainty block Δ_p is a fictitious uncertainty linking the controlled output z to the exogenous input w. This fictitious uncertainty block allows us to transform the robust performance problem in an equivalent robust stability problem with respect to the augmented structured perturbation Δ_{sa}, as shown in the figure below.



Intuitively, if the closed-loop system shown above is robustly stable to all $\|\Delta_p(s)\|_\infty < 1$, then this must mean that $\|T_{z2d}\|_\infty \le 1$ (for all admissible $\Delta_s(s)$) by virtue of the small gain theorem. This intuition is right, as shown by the following robust performance theorem. Note that the generalized plant is 2x2: $P(s) = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$, with its first input signal being the output of $\tilde{\Delta}_{sa}(s)$, and its first output signal being the input to $\tilde{\Delta}_{sa}(s)$.

Theorem: Robust performance with structured perturbation

Assume controller $K(s)$ is stabilizing for the nominal plant $P(s)$. Then for the closed-loop system in the above figure, for all $\Delta_s \in \mathcal{S}, \|\Delta_s\|_\infty < \beta$ the closed-loop system is well-posed, internally stable and $\|T_{z2d}\|_\infty \leq \frac{1}{\beta}$ if and only if

$$\sup_{\omega \in \mathbb{R}} \mu_\Omega \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\} \leq \frac{1}{\beta}, \tag{0.189}$$

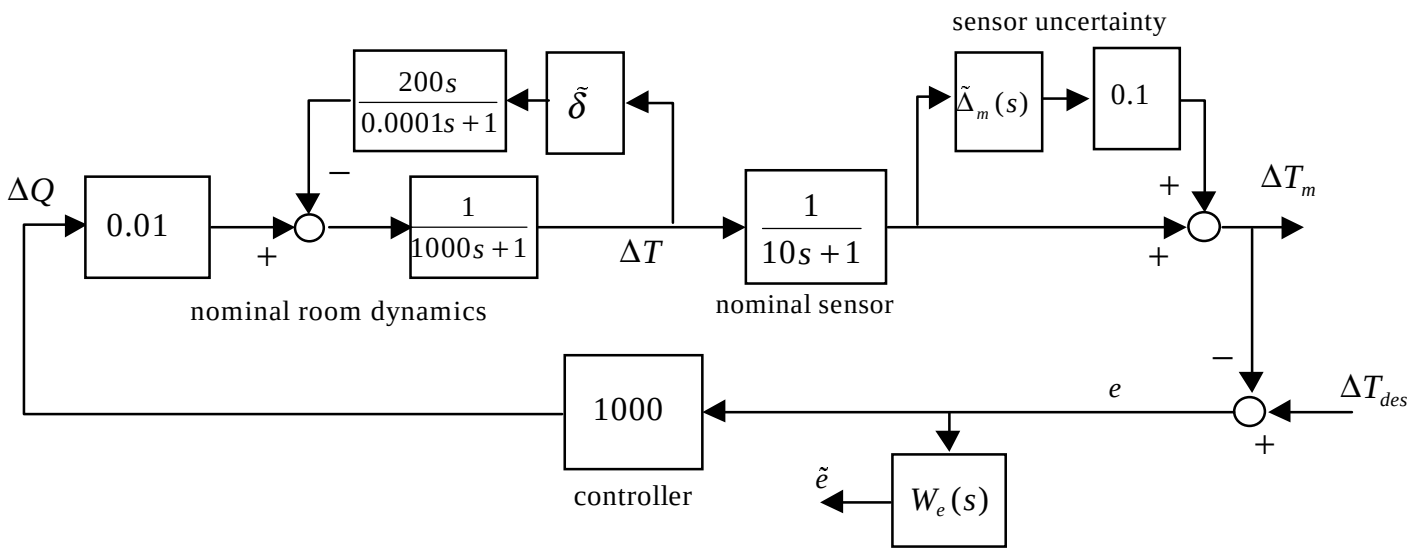
where T_{z2d} is the transfer matrix from the input d to the output z_2.

Remarks

- The theorem is used mostly with $\beta = 1$ and normalized perturbations $\tilde{\Delta}_s(s)$.
- The proof of this theorem uses the *main-loop theorem* stated above.
- This theorem allows us to check robust performance by computing $\mu_\Omega \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\}$ at "all" frequencies and making sure it is less than 1. In practice, this is done for a reasonable frequency grid of, say, 200 points. Computation of the structured singular value is readily performed in Matlab with the "mu" command.
- The condition $\sup_{\omega \in \mathbb{R}} \mu_\Omega \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\} \leq 1$ implies robust stability and nominal performance (as it should, as these are weaker specs than robust performance)
- Note that this result doesn't tell us *how* to design a controller that will achieve robust performance. It will be the subject of the next chapter.

Example: Robust performance of room temperature control system

We consider a performance spec for temperature setpoint tracking.



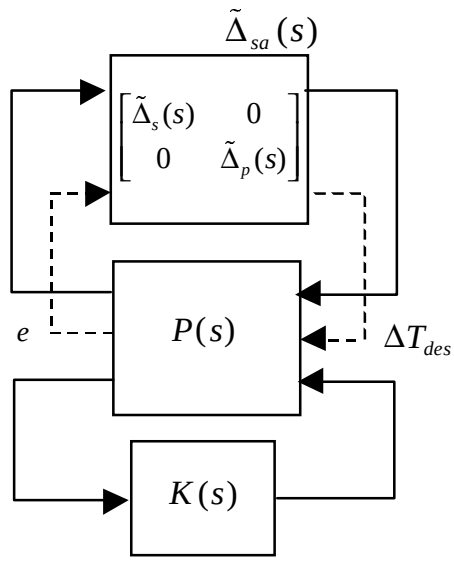
The LFT obtained from the above block diagram has the desired temperature ΔT_{des} as an input and the weighted room temperature error $\tilde{e} := W_e(s)(\Delta T_{des} - \Delta T_m)$ as the output to be minimized. Right away, we can define the augmented structured perturbation with a fictitious block for performance:

$$\tilde{\Delta}_{\mathrm{sa}}(s) := \begin{bmatrix} \tilde{\Delta}_{\mathrm{m}}(s) & 0 \\ 0 & \tilde{\Delta}_{\mathrm{p}}(s) \end{bmatrix}, \tag{0.190}$$

and define the augmented block structure:

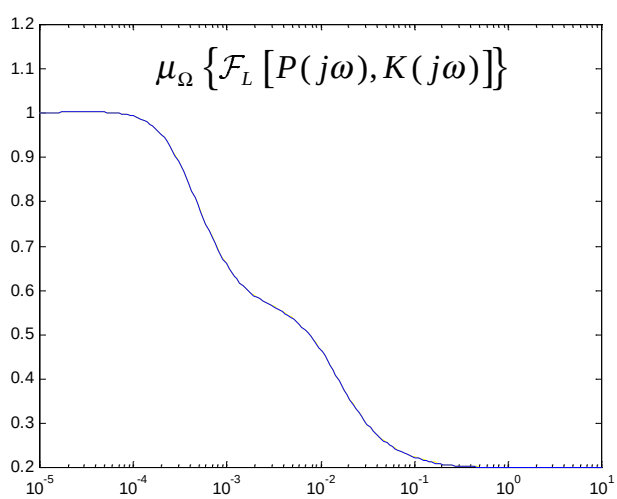
$$\Omega := \left\{ \Delta_{sa} = \begin{bmatrix} \Delta_s & 0 \\ 0 & \Delta_p \end{bmatrix} : \Delta_s \in \Gamma, \Delta_p \in \mathbb{C} \right\} \tag{0.191}$$

The block diagram above can be rearranged into an LFT form for the equivalent robust stability problem.



Note that we must lump all the input and output signals going to/coming from the augmented perturbation together, so that the generalized plant is 2x2: $P(s) = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$. We won't do this by hand, but rather use Matlab to build the LFT and compute the robust performance condition (see *muroomheating.m*).

Selecting the performance weighting function as $W_e(s) = \dfrac{10}{2500s+1}$, a plot of $\mu_\Omega\left\{\mathcal{F}_L\left[P(j\omega), K(j\omega)\right]\right\}$ is shown below indicating that robust performance was attained. Note that it often takes a few iterations in which the designer has to relax the performance weighting function (less gain, lower bandwidth) before robust performance can be achieved. Sometimes the uncertainty is just too large and robust performance can't be achieved. This is the robustness/performance tradeoff at play.

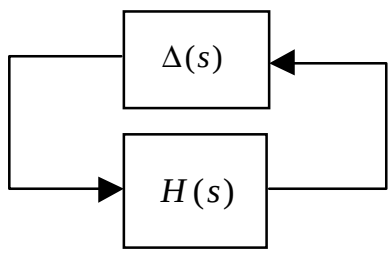


5 Robust H_∞ Control Design

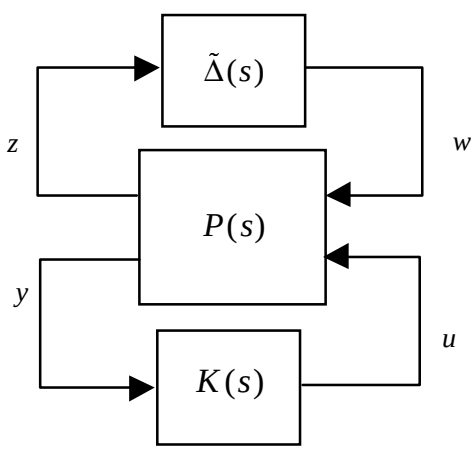
We have seen that robustness of a control system to stable perturbations of its dynamics is related to the $\mathcal{H}_\infty$-norms of certain closed-loop transfer matrices. Therefore we can use $\mathcal{H}_\infty$-optimal control theory to design robust control systems by minimizing $\mathcal{H}_\infty$-norms.

5.1 Objective

Recall that the small-gain theorem says that a feedback interconnection of a nominal closed-loop transfer matrix with a stable perturbation $\Delta(s) \in \mathcal{RH}_\infty$ is stable if the $\mathcal{H}_\infty$ norm of each is less than 1. This theorem is widely used for robust controller design because such an interconnection would be stable *for all* perturbations of norm less than 1 in the figure below:



As we have seen in the previous chapter, the closed-loop transfer matrix $H(s)$ is determined as the system "seen" by the perturbation. The transfer matrix is relatively straightforward to obtain by solving the loop equations. As previously mentioned any type of uncertainty model can be transformed into an upper LFT. In the case of $H(s)$ being a lower LFT of a generalized plant and a controller, we have the following block diagram:



The objective of controller design for robust stability with this setup is to design a finite-dimensional, linear time-invariant controller $K(s)$ such that the nominal closed-loop system $H(s) := \mathcal{F}_L\left[P(s), K(s)\right]$ is stable, and the perturbed closed-loop system $\mathcal{F}_U\left[H(s), \tilde{\Delta}(s)\right]$ is stable for all $\left\|\tilde{\Delta}(s)\right\|_\infty < 1$.

The necessary and sufficient condition to achieve robust stability with normalized linear fractional uncertainty $\tilde{\Delta}(s)$ is

$$\left\|\mathcal{F}_L\left[P(s), K(s)\right]\right\|_\infty \le 1. \tag{0.192}$$

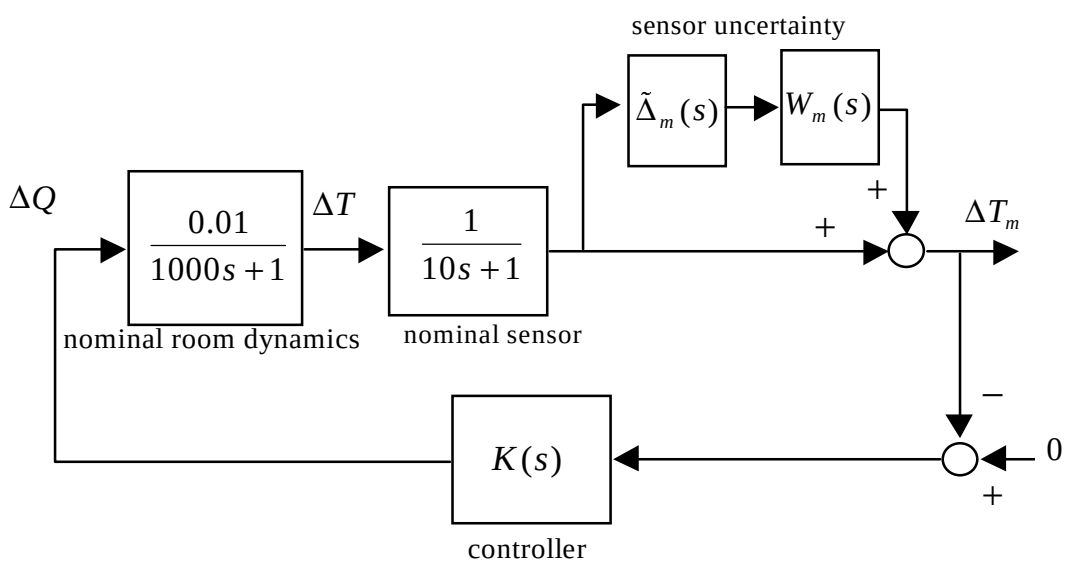
Therefore, the controller design problem for robust stability is just an $\mathcal{H}_\infty$-norm minimization problem:

$$\min_K \left\|\mathcal{F}_L\left[P(s), K(s)\right]\right\|_\infty . \tag{0.193}$$

Example: Room temperature $\mathcal{H}_\infty$ design for robust stability

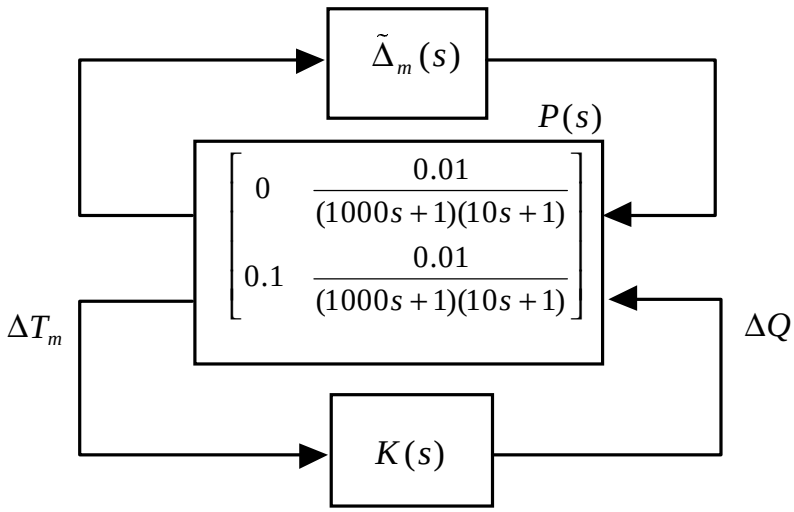
We want to maximize the stability robustness of our room temperature control system with respect to the uncertainty in sensor dynamics.

The temperature sensor with first-order dynamics was assumed to have 10% uncertainty in its frequency response, i.e., $\left\|\Delta_l(j\omega)\right\| < 0.1$, $\forall\omega$, so that $W_m(s) = 0.1$. This uncertainty was modeled with an output multiplicative model, which was then transformed into an LFT model.



The corresponding LFT is shown below, where the generalized plant $P(s)$ is

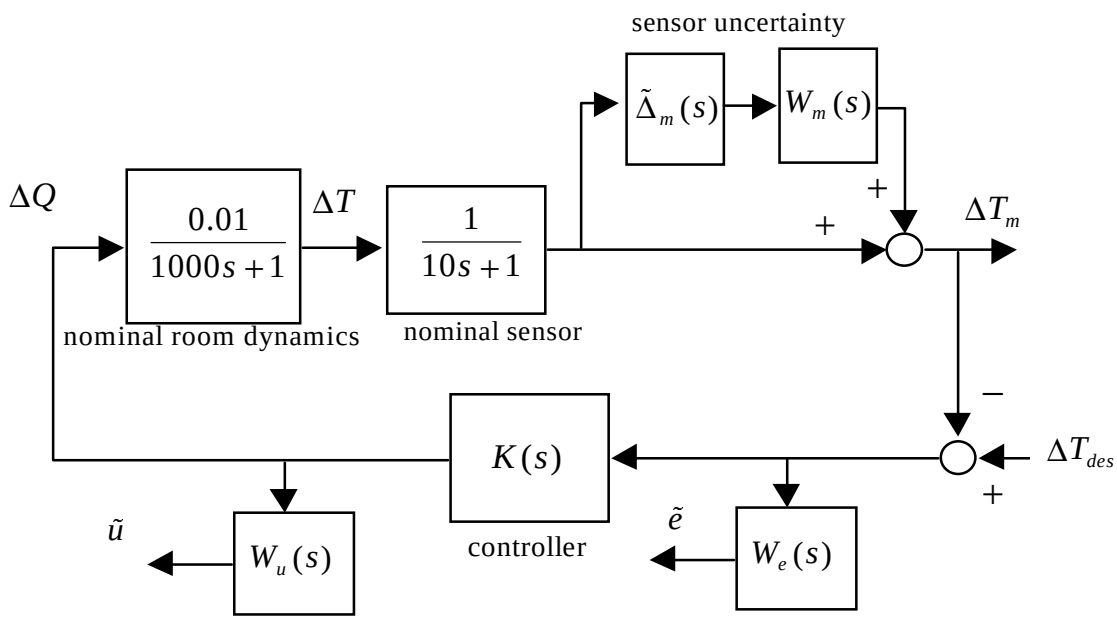
$$P_{11} = 0,\ \ P_{12} = G,\ \ P_{21} = W_m,\ \ P_{22} = G \tag{0.194}$$



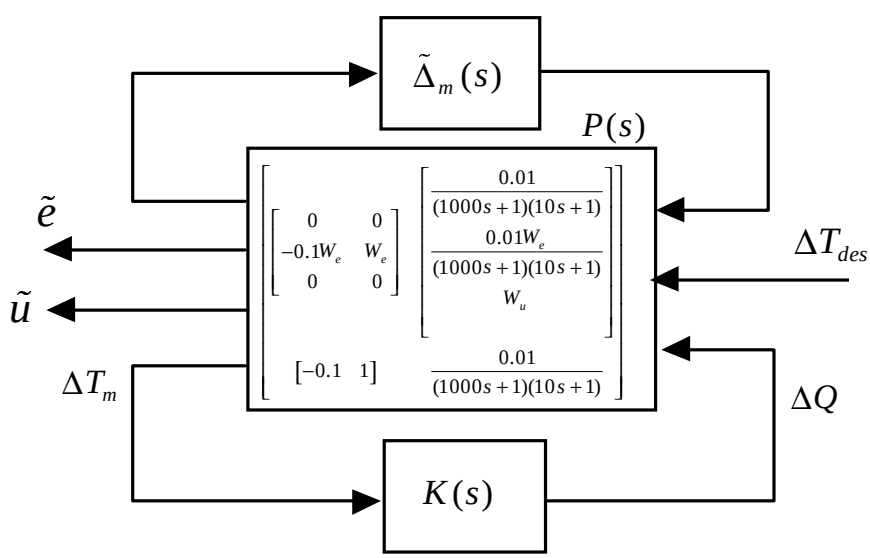
The controller is designed to minimize the $\mathcal{H}_\infty$-norm of the lower LFT:

$$\min_K \left\| \mathcal{F}_L\left[P(s), K(s)\right] \right\|_\infty . \tag{0.195}$$

It turns out that the optimally-robust controller is no controller at all, i.e., $K(s) = 0$! In this case, the lower LFT is simply 0 which is stable, and $\left\| \mathcal{F}_L\left[P(s), K(s)\right] \right\|_\infty = \left\| \mathcal{F}_L\left[P(s), 0\right] \right\|_\infty = \left\| 0 \right\|_\infty = 0$ is minimized. Then, any stable perturbation of any size can be tolerated since the resulting open-loop system will always be stable. To avoid this trivial solution, we must add a performance specification for temperature tracking with $W_e(s) = \dfrac{10}{1000s+1}$. To make the problem well-posed, we also add a weighting function on the control signal. Let's add the weighting function $W_u(s) = 0.001$.



The corresponding LFT is shown below, with the generalized plant $P(s)$.



This is done in the m-file *Hinfroomheating.m.* All of the assumptions on the generalized plant previously explained earlier for $\mathcal{H}_\infty$-optimal control design must be checked first. Here is the output produced by the *hinfsyn* function, indicating that the $\mathcal{H}_\infty$-norm of the lower LFT $\left\|\mathcal{F}_L\left[P(s),K(s)\right]\right\|_\infty$ ("gamma value achieved") is almost equal to 1. This implies that the $\mathcal{H}_\infty$-norms of sub-blocks of

$\mathcal{F}_L\left[P(s), K(s)\right]$ are all less than or equal to 1, some of them being the closed-loop transfer matrices that we wanted to minimize to get robust stability, and nominal performance. This technique is actually called the *mixed-sensitivity problem* where the $\mathcal{H}_\infty$-norm of a bigger transfer matrix than what the original robustness or performance spec called for is minimized.

Test bounds: 0.0000 < gamma <= 2.0000

gamma	hamx_eig	xinf_eig	hamy_eig	yinf_eig	nrho_xy	p/f
2.000	8.7e-003	2.0e-003	1.0e-003	-3.0e-021	0.0000	p
1.000	2.5e-016#	*******	1.0e-003	-1.3e-042	******	f
1.500	7.5e-003	2.0e-003	1.0e-003	1.5e-042	0.0000	p
1.250	6.0e-003	2.0e-003	1.0e-003	-1.8e-021	0.0000	p
1.125	4.6e-003	2.0e-003	1.0e-003	-1.0e-041	0.0000	p
1.063	3.4e-003	2.0e-003	1.0e-003	-1.1e-021	0.0000	p
1.031	2.5e-003	2.0e-003	1.0e-003	-9.5e-022	0.0000	p
1.016	1.8e-003	2.0e-003	1.0e-003	4.2e-042	0.0000	p
1.008	1.2e-003	2.0e-003	1.0e-003	6.1e-042	0.0000	p
1.004	8.9e-004	2.0e-003	1.0e-003	5.8e-041	0.0000	p
1.002	6.3e-004	2.0e-003	1.0e-003	-2.5e-041	0.0000	p
1.001	4.4e-004	2.0e-003	1.0e-003	-5.7e-022	0.0000	p

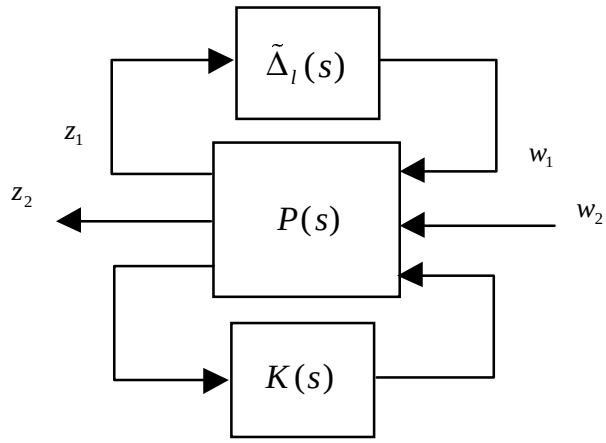
Gamma value achieved: 1.0010

The robust $\mathcal{H}_\infty$ controller obtained is of third order.

5.2 Mixed-sensitivity robust H_∞ controller design

The last example dealt with two concurrent objectives: maximize the robust stability margin, and minimize the sensitivity. These two objectives lead to the minimization of the $\mathcal{H}_\infty$-norms of two different transfer matrices. But $\mathcal{H}_\infty$ control can only deal with *one objective at a time*, e.g., a single full uncertainty block, or a single performance specification.

One approach to satisfy two (or more) objectives at the same time is to simply add inputs and outputs corresponding to all of these objectives to the generalized plant, and then minimize the $\mathcal{H}_\infty$-norm of the big closed-loop transfer matrix (lower LFT) relating all of the inputs to all of the outputs. This is what we did in the previous room temperature control example. It is called a *mixed-sensitivity design*. If this $\mathcal{H}_\infty$-norm is less than or equal to 1, then all of the objectives are attained as a result (it is a sufficient condition). Consider for example the block diagram below.



If we can somehow manage to design a controller that will give

$$\left\| \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \mapsto \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \right\|_\infty = \left\| \mathcal{F}_L \left[P(s), K(s) \right] \right\|_\infty \leq 1, \tag{0.196}$$

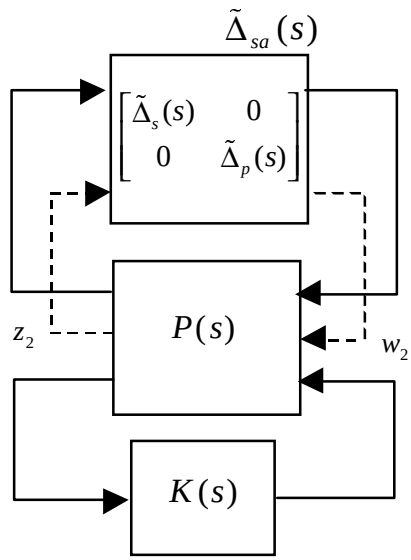
then as a result we will get $\left\| T_{z_1 w_1} \right\|_\infty \leq 1$ (robust stability), and $\left\| T_{z_2 w_2} \right\|_\infty \leq 1$ (nominal performance). However, this is a conservative approach, because $\mathcal{F}_L \left[P(s), K(s) \right]$ contains other transfer matrices that are minimized as well, putting more constraints than called for. The μ-synthesis approach for controller design is often a better solution.

6 Controller Design Via Mu-Synthesis

We now present the μ -synthesis technique to design controllers for robust performance. But first, let's review the main result of μ -analysis. Consider the perturbed closed-loop system in the figure below, where $z_2(t) \in \mathbb{R}^{n_{z2}}$ and $w_2(t) \in \mathbb{R}^{n_{w2}}$, and define the augmented block structure

$$\Omega := \left\{ \Delta_{sa} = \begin{bmatrix} \Delta_s & 0 \\ 0 & \Delta_p \end{bmatrix} : \Delta_s \in \Gamma,\ \Delta_p \in \mathbb{C}^{n_{w2} \times n_{z2}} \right\} \tag{0.197}$$

The uncertainty block Δ_p is a fictitious uncertainty linking the controlled output z_2 to the exogenous input w_2.



Recall the theorem on robust performance with structured perturbation

Assume controller $K(s)$ is stabilizing for the nominal plant $P(s)$. Then for the above block diagram, for all $\Delta_s \in \mathcal{S}, \|\Delta_s\|_\infty < 1$ the closed-loop system is well-posed, internally stable and $\|T_{z2d}\|_\infty \le 1$ if and only if $\sup_{\omega \in \mathbb{R}} \mu_\Omega \left\{ \mathcal{F}_L \left[P(j\omega), K(j\omega) \right] \right\} \le 1$.

A μ *-synthesis* is the solution to the following optimization problem:

$$\min_{K(s)\text{ stabilizing}} \sup_{\omega\in\mathbb{R}} \mu_\Omega \left\{\mathcal{F}_L\left[P(j\omega),K(j\omega)\right]\right\} \tag{0.198}$$

This is still an open problem. Recall that an upper bound on the structured singular value $\mu_\Omega \left\{\mathcal{F}_L\left[P(j\omega),K(j\omega)\right]\right\}$ is given by $\inf_{D\in\mathcal{D}} \bar{\sigma}\left(DMD^{-1}\right)$

$$\mathcal{D}:=\left\{\begin{array}{c}\operatorname{diag}\left\{D_1,\ldots,D_s,d_1 I_{m_1},\ldots,d_{F-1}I_{m_{F-1}},I_{m_F}\right\}: \\ D_i\in\mathbb{C}^{r_i x r_i}, D_i=D_i^*>0, d_j\in\mathbb{R}, d_j>0\end{array}\right\}. \tag{0.199}$$

which is the set of matrices that commute with any $\Delta_s \in \Gamma$.

However, a reasonable approach is to minimize the upper bound on $\mathcal{F}_L\left[P(j\omega),K(j\omega)\right]$, i.e., to "solve"

$$\min_{K(s)} \inf_{D(s),D^{-1}(s)\in\mathcal{RH}_\infty} \left\|D(s)\mathcal{F}_L\left(P(s),K(s)\right)D^{-1}(s)\right\|_\infty \tag{0.200}$$

by iteratively solving for the controller $K(s)$ and the D-scale $D(s)$. This is the so-called *D-K Iteration.* The stable and minimum phase scaling matrix $D(s)$ is chosen such that $D(s)\Delta(s)=\Delta(s)D(s)$ (note that *D*(*s*) is not necessarily belong to $\mathbf{D}$ since *D*(*s*) is not necessarily Hermitian).

For a fixed scaling transfer matrix $D(s)$,

$$\min_{K(s)} \left\|D\mathcal{F}_L\left(G,K\right)D^{-1}\right\|_\infty \tag{0.201}$$

is a standard H_∞ optimization problem. For a given stabilizing controller $K(s)$,

$$\inf_{D,D^{-1}\in\mathcal{H}_\infty} \left\|D\mathcal{F}_L\left(G,K\right)D^{-1}\right\|_\infty \tag{0.202}$$

is a standard convex optimization problem and it can be solved pointwise in the frequency domain:

$$\sup_\omega \inf_{D_\omega\in\mathcal{D}} \bar{\sigma}\left[D_\omega F_l\left(G,K\right)\left(j\omega\right)D_\omega^{-1}\right] \tag{0.203}$$

Indeed,

$$\inf_{D,D^{-1}\in\mathcal{RH}_\infty}\left\|D\mathcal{F}_L(G,K)D^{-1}\right\|_\infty=\sup_\omega\inf_{D_\omega\in\mathcal{D}}\bar{\sigma}\left[D_\omega\mathcal{F}_L(G,K)(j\omega)D_\omega^{-1}\right]. \quad (0.204)$$

This last equation follows intuitively from the following arguments: the left-hand side is always no smaller than the right-hand side, and, on the other hand, given the minimizing D_ω from the right hand side across the frequency, there is always a rational function $D(s)$ uniformly approximating the magnitude frequency response D_ω.

Note that when S=0, (no scalar blocks, only full bocks)

$$D_\omega=diag\left(d_1^\omega I,\ldots,d_{F-1}^\omega I,I\right)\in\mathcal{D}, \quad (0.205)$$

which is a block-diagonal scaling matrix applied pointwise across frequency to the frequency response $\mathcal{F}_L(G,K)(j\omega)$.

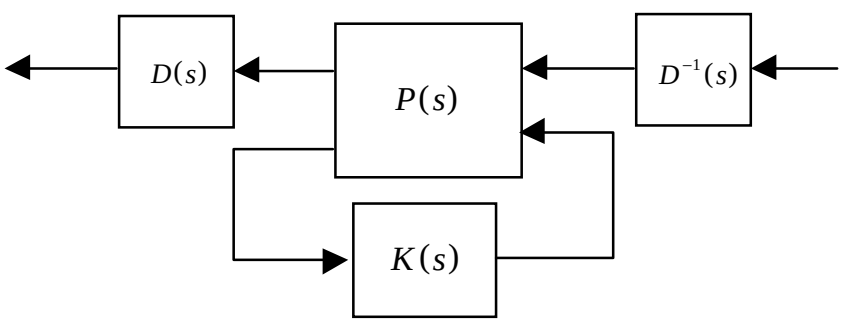


Figure 32: μ -synthesis via scaling

D-K Iterations proceed by performing this two-parameter minimization in sequential fashion: first minimizing over $K(s)$ with $D(s)$ fixed, then minimizing pointwise over D_ω with $K(s)$ fixed, then again over $K(s)$, and again over D_ω, etc. Details of this process are summarized in the following steps:

1. Fix an initial estimate of the scaling matrix $D_\omega\in\mathcal{D}$ pointwise across frequency.

2. Find scalar transfer functions $d_i(s),d_i^{-1}(s)\in\mathcal{RH}_\infty$ for $i=1,\ldots,F-1$ such that $\left|d_i(j\omega)\right|\approx d_i^\omega$. This interpolation process is currently done mostly by graphical matching using lower order transfer functions.

3. Let $D(s) = \mathrm{diag}\{d_1(s)I, \ldots, d_{F-1}(s)I, I\}$. Construct a state space model for the system in the above figure:

$$\hat{P}(s) = \begin{bmatrix} D(s) & 0 \\ 0 & I \end{bmatrix} P(s) \begin{bmatrix} D^{-1}(s) & 0 \\ 0 & I \end{bmatrix}. \tag{0.206}$$

4. Solve an $\mathcal{H}_\infty$-optimization problem to minimize $\left\| \mathcal{F}_L\left(\hat{P}, K\right) \right\|_\infty$ over all stabilizing controllers. Note that this optimization problem uses the scaled version of $P(s)$. Let its minimizing controller be denoted by $\hat{K}$.

5. Minimize $\bar{\sigma}\left[D_\omega \mathcal{F}_L\left(P, \hat{K}\right) D_\omega^{-1} \right]$ over D_ω, pointwise across frequency. Note that this evaluation uses the minimizing $\hat{K}$ from the last step, but that $P(s)$ is *unscaled*. The minimization itself produces a new scaling function. Let this new function be denoted by $\hat{D}_\omega$.

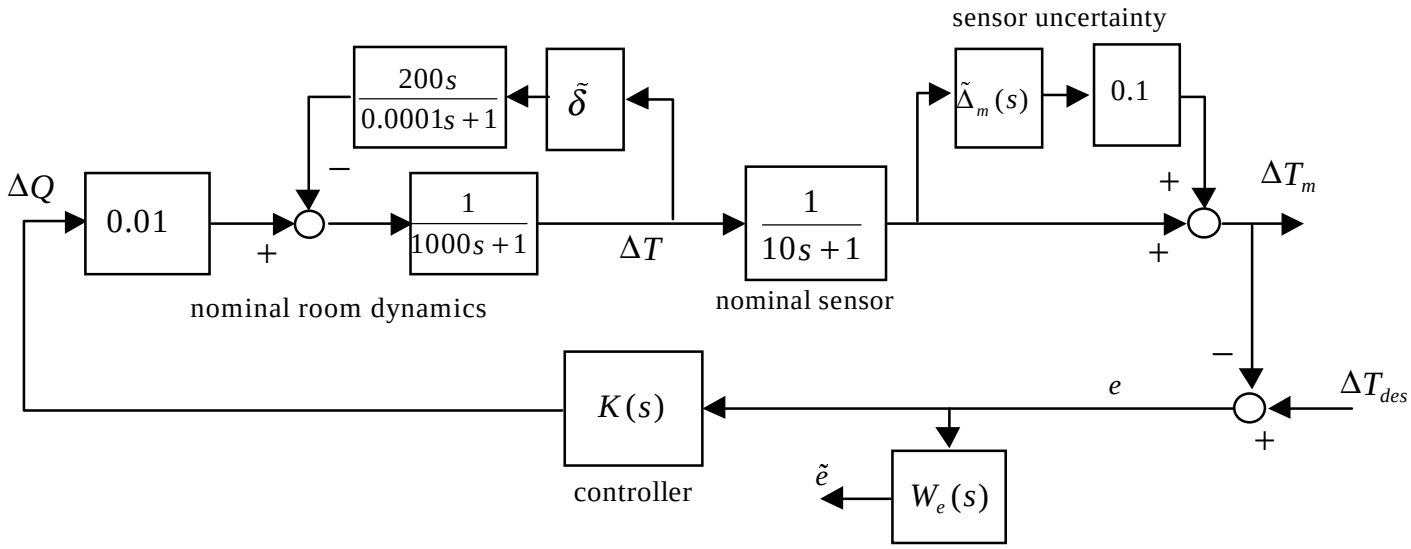
6. Compare $\hat{D}_\omega$ with the previous estimate D_ω. Stop if they are close, but, otherwise, replace D_ω with $\hat{D}_\omega$ and return to step (2).

With either $K(s)$ or $D(s)$ fixed, the global optimum in the other variable may be found using the μ and $\mathcal{H}_\infty$ solutions. Although the joint optimization over $K(s)$ and $D(s)$ is not convex and the global convergence is not guaranteed, many designs have shown that this approach works very well. In fact, this is probably the most effective design methodology available today for dealing with such complicated multivariable problems.

We will solve two examples to show how powerful the μ-synthesis technique is. The *dkit* is used to implement the DK-iteration.

Example: μ -synthesis for robust performance of room temperature control system

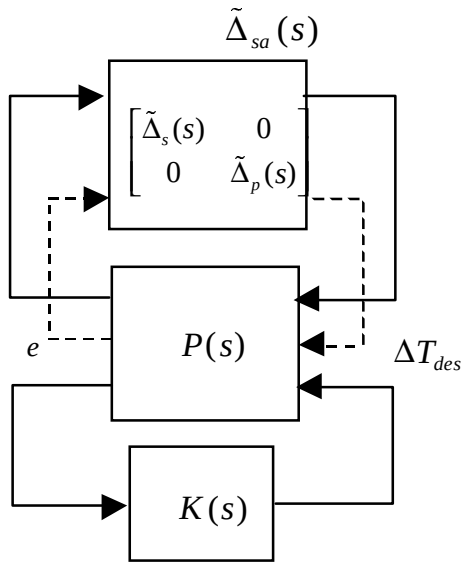
We consider a performance spec for temperature setpoint tracking.



The LFT obtained from the above block diagram has the desired temperature ΔT_{des} as an input and the weighted room temperature error $\tilde{e} := W_e(s)(\Delta T_{des} - \Delta T_m)$ as the output to be minimized. Recall that we defined the augmented structured perturbation with a fictitious block for performance: $\tilde{\Delta}_{sa}(s) := \begin{bmatrix} \tilde{\Delta}_m(s) & 0 \\ 0 & \tilde{\Delta}_p(s) \end{bmatrix}$, and we also defined the augmented block structure:

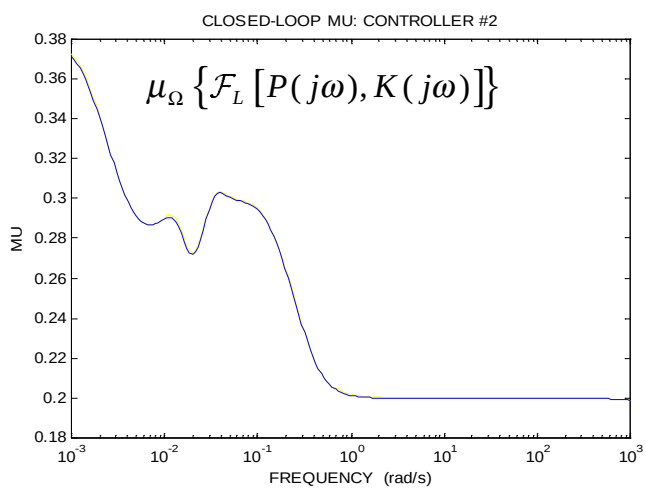
$$\Omega := \left\{ \Delta_{sa} = \begin{bmatrix} \Delta_s & 0 \\ 0 & \Delta_p \end{bmatrix} : \Delta_s \in \Gamma,\ \Delta_p \in \mathbb{C} \right\}. \tag{0.207}$$

The block diagram above can be rearranged into an LFT form for the equivalent robust stability problem.



Note that we must lump all the input and output signals going to/coming from the augmented perturbation together, so that the generalized plant is 2x2: $P(s) = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$. We use Matlab to build the LFT and implement a D-K iteration. (see *musynroomheating.m*).

The performance weighting function is $W_e(s) = \dfrac{10}{2500s+1}$. A plot of $\mu_\Omega\left\{\mathcal{F}_L\left[P(j\omega), K(j\omega)\right]\right\}$ is shown below indicating that robust performance was attained. Note that there is room to modify the performance weighting function (more gain, higher bandwidth) for better robust performance, since $\mu_\Omega\left\{\mathcal{F}_L\left[P(j\omega), K(j\omega)\right]\right\}$ is well below 1.



The robust controller obtained is of the fourth order. Its state-space matrices as given by Matlab are shown below:

```
A matrix
 1.0e+003 *

  -0.0243  -0.0082   0.0016   0.0002
   0.0703   0.0236  -0.0046  -0.0006
   0.0044   0.0015  -0.0012  -0.0763
   0.0000   0.0033  -0.1079  -9.8977

B matrix
   0.0374
   0.0015
  -0.0098
   0.0001

C matrix
 1.0e+004 *

   9.1902   3.0789  -0.6032  -0.0723

D matrix
    0
```

Example: μ -synthesis for robust performance of mixing tank process control system

Recall that the nominal state-space representation of the mixing tank process is:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{0.208}$$

where

$$A = \begin{bmatrix} 0 & 0 \\ 0.0298 & -0.001 \end{bmatrix} B = \begin{bmatrix} 1 & 0 \\ -30 & 2.38\times10^{-7} \end{bmatrix} C = \begin{bmatrix} 0 & 1 \end{bmatrix} D = 0_{2\times2}. \tag{0.209}$$

We consider an uncertainty in the parameter T_{in}^{*}, i.e., the assumed operating liquid temperature at the inlet valve. Variations in T_{in}^{*} are evaluated to be within $\pm$5% of the nominal temperature. In fact we

can write T_{in}^{*} as : $T_{in}^{*} = 293 \pm 15$. This variation in the temperature T_{in}^{*} will lead to variation in the matrices representing the state-space model. If we take the extreme values of T_{in}^{*}, we can compute the extreme values of the state space matrices:

For $T_{in}^{*} = 293+15$ $A = \begin{bmatrix} 0 & 0 \\ 0.0148 & -0.001 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 \\ -15 & 2.38\times10^{-7} \end{bmatrix}$ $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$ $D = 0_{2\times2}$.

For $T_{in}^{*} = 293-15$ $A = \begin{bmatrix} 0 & 0 \\ 0.0448 & -0.001 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 \\ -45 & 2.38\times10^{-7} \end{bmatrix}$ $C = \begin{bmatrix} 0 & 1 \end{bmatrix}$ $D = 0_{2\times2}$.

In the framework of the uncertainty representation the perturbed matrices can be written as:

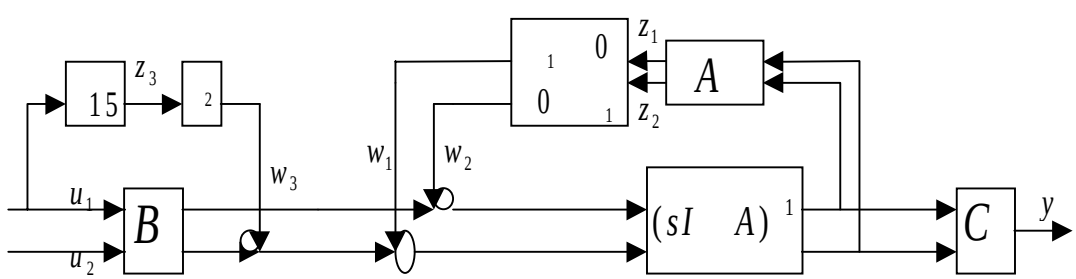
$$A_p = \begin{bmatrix} 0 & 0 \\ 0.0298 & -0.001 \end{bmatrix} + \begin{bmatrix} \tilde{\delta}_1 & 0 \\ 0 & \tilde{\delta}_1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0.015 & 0 \end{bmatrix} \tag{0.210}$$

and

$$B_p = \begin{bmatrix} 1 & 0 \\ -30 & 2.38\times10^{-7} \end{bmatrix} + \delta_2 \begin{bmatrix} 0 & 0 \\ -15 & 0 \end{bmatrix} \tag{0.211}$$

such that $-1 < \delta_1, \delta_2 < 1$.

We can now represent the uncertain transfer matrix of the process as given in the figure below:



Where $A_{\Delta} := \begin{bmatrix} 0 & 0 \\ 0.015 & 0 \end{bmatrix}$.

In order to convert the design problem for this uncertain plant into LFT representation, we need to find the generalized plant $P(s)$ that includes the outputs of the uncertainty parameters w_1, w_2, w_3 as exogenous inputs and the inputs of the parameters uncertainty as variables to control z_1, z_2, z_3. This is tedious to do by hand, but is easy to do in Matlab. See the m-file *musynmixing.m* given in appendix.

7 Acronyms

LTI: Linear time-invariant

IMC: Internal model control

MIMO: Multi-input multi-output

SISO: Single-input single-output

LFT: Linear fractional transformation

8 References

[1] K. Zhou, J.C. Doyle, K. Glover
"Robust and Optimal Control", Prentice Hall, 1996, 596 pp.

[2] K. Zhou, J.C. Doyle
"Essentials of Robust Control", Prentice Hall, 1998, 411 pp.

[3] S. Skogestad, I. Postlethwaite
"Multivariable Feedback Control", Wiley, 1996,559 pp.

[4] M. Morari, E. Zafiriou,
"Robust Process Control", Prentice Hall, 1989.

9 Appendix: Matlab m-files

H2mixing.m

```
%% (c) Coronado Systems 2000
% Author: Benoit Boulet
% This m-file was used to compute and simulate the optimal H-infinity
controller
% for the mixing tank process

% 2-input, 1-output plant state-space matrices
A=[0 0; 0.0298 -0.001];
B=[1 0; -30 2.388e-7];
C=[0 1];         % only the temperature sensor is used
D=zeros(1,2);
G=pck(A,B,C,D);

% weighting functions
numWo=[0.01 10];
denWo=[2000 1];
Wo=nd2sys(numWo,denWo);

numWi1=[0.001];
denWi1=[1];
Wi1=nd2sys(numWi1,denWi1);
Wi=daug(Wi1,Wi1);

numWe=[10];
denWe=[2000 1];
We=nd2sys(numWe,denWe);

numWu1=[10];
denWu1=[1];
numWu2=[0.0001];
denWu2=[1];
Wu1=nd2sys(numWu1,denWu1);
Wu2=nd2sys(numWu2,denWu2);
Wu=daug(Wu1,Wu2);

% Build the system interconnection to obtain the generalized plant P(s)
% using the "sysic" command

systemnames=' Wo Wi Wu We G ';
inputvar='[ w1; w2{2}; u{2} ]';
outputvar='[ Wu; We; Wo - G ]';
input_to_Wo='[ w1 ]';
input_to_Wi='[ w2 ]';
input_to_Wu='[ u ]';
input_to_We='[ Wo - G ]';
input_to_G='[ u + Wi ]';
sysoutname='P';
```

```
cleanupsysic='yes';
sysic;

% Extract partitioned state-space matrices
[AP,BP,CP,DP]=unpck(P);
BP1=BP(:,1:3);
BP2=BP(:,4:5);
CP1=CP(1:3,:);
CP2=CP(4,:);
DP11=DP(1:3,1:3);
DP12=DP(1:3,4:5);
DP21=DP(4,1:3);
DP22=DP(4,4:5);


% Stabilizability of (AP,BP2) and (AP,BP1)
% PBH test: [-AP+sI BPi] must not lose rank at closed RHP eigvals of AP
rhpeigs=[];
[V,EIGS] = eig(AP);
eigs = diag(EIGS);

for i=1:length(AP)
if real(eigs(i)) >= 0
      rhpeigs=[rhpeigs;eigs(i)];
end
end

p = length(rhpeigs);
r = length(AP);      % Rank of [sI-AP] for s not eigval(AP)
rkb1=[];             % ranks at rhp eigenvalues
rkb2=[];             % ranks at rhp eigenvalue

for j=1:p
rkb1=[rkb1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP1])];
rkb2=[rkb2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP2])];
end

if min(rkb1) < length(AP)
  sprintf('PROBLEM: (AP,BP1) IS NOT STABILIZABLE')
  stop
end

if min(rkb2) < length(AP)
  sprintf('PROBLEM: (AP,BP2) IS NOT STABILIZABLE')
  stop
end

% Detectability of (CP2,AP) and (CP1,AP)
% PHB test: [(-AP+sI)' CP2']' must not lose rank at closed RHP eigvals
of AP

rkc1=[];              % ranks at rhp eigenvalues
rkc2=[];              % ranks at rhp eigenvalues

for j=1:p
   rkc1=[rkc1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP1])];
```

```
   rkc2=[rkc2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP2])];
end

if min(rkc1) < length(AP)
  sprintf('PROBLEM: (CP1,AP) IS NOT DETECTABLE')
  stop
end

if min(rkc2) < length(AP)
  sprintf('PROBLEM: (CP2,AP) IS NOT DETECTABLE')
  stop
end

% There are two more condition that should be checked. The matrices
% [AP-jwI BP2; CP1 DP12] and [AP-jwI BP1; CP2 DP21] must have full
column
% and row rank respectively for all w.

ww  = logspace(-4,2,200);

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP2; CP1 DP12]);
  if rk < (length(AP)+length(BP2(1,:)))
     sprintf('PROBLEM: FIRST MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP1; CP2 DP21]);
  if rk < (length(AP)+length(CP2(:,1)))
     sprintf('PROBLEM: SECOND MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end


% compute the H2 optimal controller
nmeas=1; % number of measurements to controller
ncon=2;  % number of control signals from controller
ricmethod=2;
[K,Tzw,norms,Kfi,Gfi,hamx,hamy]=h2syn(P,nmeas,ncon,ricmethod)

% H2-norm achieved
h2norm(Tzw)

% closed-loop sensitivity from output temperature disturbance to
process output temp.
% i.e. S(s)=(I+GK)^(-1)
S=starp(abv(sbs(0,1),sbs(1,mmult(-1,G,K))),1);

% closed-loop system from output temperature disturbance to process
inputs
% i.e. SK(s)=-(I-GK)^(-1)K
```

```
KS=starp(abv(sbs([0; 0],eye(2)),sbs(1,G)),mmult(K,-1));

% simulate the output response to a change in setpoint for deltaT
tfinal=10000; % final time for simulation
intstep=5;   % integration step
step_input=-1; % step input: -1K disturbance in temperature

% simulate plant response first
plant_output_resp=trsp(mmult(S,Wo,0.01),step_input,tfinal,intstep);
K_output_resp=trsp(mmult(KS,Wo,0.01),step_input,tfinal,intstep);

% plot the output and actuator responses
figure(1)
subplot(311)
vplot(plant_output_resp);
subplot(312)
vplot(sel(K_output_resp,1,1));
subplot(313)
vplot(sel(K_output_resp,2,1));
```

bodeH2IMC.m

```
%% (c) Coronado systems 2000
% This m-file was used to generate the Bode plot of
% sensitivity with the SISO IMC H2 controller design

% frequency grid
ww=logspace(-5,1,200);

% Output sensitivity transfer function
numTyd=[1.007 1339.3 0.67];
denTyd=conv([1007 1],[0.001 1]);
Tyd=nd2sys(numTyd,denTyd);

% Frequency response and Bode plot
Tydjw=frsp(Tyd,ww);

figure(1)
vplot('bode_g',Tydjw)

% Response to step disturbance
Tydstep=trsp(Tyd,5,5000,1,0);

% Plot the response
figure(2)
vplot(Tydstep)
```

Hinfmixing.m

```
%% (c) Coronado Systems 2000

% Author: Benoit Boulet
% This m-file was used to compute and simulate the optimal H-infinity
controller
% for the mixing tank process

% 2-input, 1-output plant state-space matrices
A=[0 0; 0.0298 -0.001];
B=[1 0; -30 2.388e-7];
C=[0 1];         % only the temperature sensor is used
D=zeros(1,2);
G=pck(A,B,C,D);

% weighting functions
numWo=[0.01 10];
denWo=[2000 1];
Wo=nd2sys(numWo,denWo);

numWi1=[0.001];
denWi1=[1];
Wi1=nd2sys(numWi1,denWi1);
Wi=daug(Wi1,Wi1);

numWe=[10];
denWe=[2500 1];
We=nd2sys(numWe,denWe);

numWu1=[10];
denWu1=[1];
numWu2=[0.0001];
denWu2=[1];
Wu1=nd2sys(numWu1,denWu1);
Wu2=nd2sys(numWu2,denWu2);
Wu=daug(Wu1,Wu2);

% Build the system interconnection to obtain the generalized plant P(s)
% using the "sysic" command
systemnames=' Wo Wi Wu We G ';
inputvar='[ w1; w2{2}; u{2} ]';
outputvar='[ Wu; We; Wo - G ]';
input_to_Wo='[ w1 ]';
input_to_Wi='[ w2 ]';
input_to_Wu='[ u ]';
input_to_We='[ Wo - G ]';
input_to_G='[ u + Wi ]';
sysoutname='P';
cleanupsysic='yes';
sysic;

% Extract partitioned state-space matrices
[AP,BP,CP,DP]=unpck(P);
BP1=BP(:,1:3);
```

```
BP2=BP(:,4:5);
CP1=CP(1:3,:);
CP2=CP(4,:);
DP11=DP(1:3,1:3);
DP12=DP(1:3,4:5);
DP21=DP(4,1:3);
DP22=DP(4,4:5);

% Stabilizability of (AP,BP2) and (AP,BP1)
% PBH test: [-AP+sI BPi] must not lose rank at closed RHP eigvals of AP
rhpeigs=[];
[V,EIGS] = eig(AP);
eigs = diag(EIGS);

for i=1:length(AP)
if real(eigs(i)) >= 0
      rhpeigs=[rhpeigs;eigs(i)];
end
end

p = length(rhpeigs);
r = length(AP);      % Rank of [sI-AP] for s not eigval(AP)
rkb1=[];             % ranks at rhp eigenvalues
rkb2=[];             % ranks at rhp eigenvalue

for j=1:p
rkb1=[rkb1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP1])];
rkb2=[rkb2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP2])];
end

%if min(rkb1) < length(AP)
%  sprintf('PROBLEM: (AP,BP1) IS NOT STABILIZABLE')
%  stop
%end

if min(rkb2) < length(AP)
  sprintf('PROBLEM: (AP,BP2) IS NOT STABILIZABLE')
  stop
end

% Detectability of (CP2,AP) and (CP1,AP)
% PHB test: [(-AP+sI)' CP2']' must not lose rank at closed RHP eigvals
of AP

rkc1=[];              % ranks at rhp eigenvalues
rkc2=[];              % ranks at rhp eigenvalues

for j=1:p
   rkc1=[rkc1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP1])];
   rkc2=[rkc2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP2])];
end

if min(rkc1) < length(AP)
  sprintf('PROBLEM: (CP1,AP) IS NOT DETECTABLE')
  stop
end
```

```
if min(rkc2) < length(AP)
  sprintf('PROBLEM: (CP2,AP) IS NOT DETECTABLE')
  stop
end

% There are two more condition that should be checked. The matrices
% [AP-jwI BP2; CP1 DP12] and [AP-jwI BP1; CP2 DP21] must have full
column
% and row rank respectively for all w.

ww  = logspace(-4,2,200);

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP2; CP1 DP12]);
  if rk < (length(AP)+length(BP2(1,:)))
     sprintf('PROBLEM: FIRST MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP1; CP2 DP21]);
  if rk < (length(AP)+length(CP2(:,1)))
     sprintf('PROBLEM: SECOND MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end


% compute the H-infinity optimal controller
nmeas=1; % number of measurements to controller
ncon=2;  % number of control signals from controller
ricmethod=2;
[K,Tzw,norms,X,Y,hamx,hamy]=hinfsyn(P,nmeas,ncon,0,1000,ricmethod);

% Hinf norm achieved
hinfnorm(Tzw)

% closed-loop sensitivity from output temperature disturbance to
process output temp.
% i.e. S(s)=(I+GK)^(-1)
S=starp(abv(sbs(0,1),sbs(1,mmult(-1,G,K))),1);

% closed-loop system from output temperature disturbance to process
inputs
% i.e. SK(s)=-(I-GK)^(-1)K
KS=starp(abv(sbs([0; 0],eye(2)),sbs(1,G)),mmult(K,-1));

% simulate the output response to a change in setpoint for deltaT
tfinal=10000; % final time for simulation
intstep=5;   % integration step
step_input=-1; % step input: 2K disturbance in temperature
```

```
% simulate plant response first
plant_output_resp=trsp(mmult(S,Wo,0.01),step_input,tfinal,intstep);
K_output_resp=trsp(mmult(KS,Wo,0.01),step_input,tfinal,intstep);

% plot the state responses
figure(1)
subplot(311)
vplot(plant_output_resp);
subplot(312)
vplot(sel(K_output_resp,1,1));
subplot(313)
vplot(sel(K_output_resp,2,1));
```

Hinfroomheating.m

```
%% (c) Coronado Systems 2000

% This m-file was used to generate the H-infinity controller for robust
control
% of the process model with perturbed sensor dynamics

% frequency grid
ww=logspace(-5,1,200);

% nominal plant transfer function
numG=0.01;
denG=[1000 1];
G=nd2sys(numG,denG);

% nominal sensor transfer function
numGs=1;
denGs=[10 1];
Gs=nd2sys(numGs,denGs);

% multiplicative weighting function
numWm=0.1;
denWm=1;
Wm=nd2sys(numWm,denWm);

% biproper weighting function on control signal (almost always
required)
numWu=0.00001;
denWu=1;
Wu=nd2sys(numWu,denWu);

% weighting function on error
numWe=[10];
denWe=[1000 1];
We=nd2sys(numWe,denWe);


% frequency response of nominal process model (with nominal sensor)
Gjw=frsp(mmult(G,Gs),ww);

% Build the system interconnection to obtain the generalized plant P(s)
% using the "sysic" command
nmeas=1; % number of measurements to controller
ncon=1;  % number of control signals from controller

P11=0;
P12=0;
P13=mmult(G,Gs);
P21=mmult(-1,Wm,We);
P22=We;
P23=mmult(G,Gs,We);
P31=0;
```

```
P32=0;
P33=Wu;
P41=mmult(-1,Wm);
P42=1;
P43=mmult(G,Gs);


systemnames=' P11 P12 P13 P21 P22 P23 P31 P32 P33 P41 P42 P43 ';
inputvar='[ w1; r; u ]';
outputvar='[ P11 + P12 + P13; P21 + P22 + P23; P31 + P32 + P33; P41 +
P42 + P43 ]';
input_to_P11='[ w1 ]';
input_to_P12='[ r ]';
input_to_P13='[ u ]';
input_to_P21='[ w1 ]';
input_to_P22='[ r ]';
input_to_P23='[ u ]';
input_to_P31='[ w1 ]';
input_to_P32='[ r ]';
input_to_P33='[ u ]';
input_to_P41='[ w1 ]';
input_to_P42='[ r ]';
input_to_P43='[ u ]';

sysoutname='P1';
cleanupsysic='yes';
sysic;

% get a minimal balanced realization of P
[P,sig]=sysbal(P1,1e-10)

% Extract partitioned state-space matrices
[AP,BP,CP,DP]=unpck(P);
BP1=BP(:,1:2);
BP2=BP(:,3);
CP1=CP(1:3,:);
CP2=CP(4,:);
DP11=DP(1:3,1:2);
DP12=DP(1:3,3);
DP21=DP(4,1:2);
DP22=DP(4,3);

% Stabilizability of (AP,BP2) and (AP,BP1)
% PBH test: [-AP+sI BPi] must not lose rank at closed RHP eigvals of AP
rhpeigs=[];
[V,EIGS] = eig(AP);
eigs = diag(EIGS);

for i=1:length(AP)
if real(eigs(i)) >= 0
      rhpeigs=[rhpeigs;eigs(i)];
end
end

p = length(rhpeigs);
r = length(AP);      % Rank of [sI-AP] for s not eigval(AP)
```

```
rkb1=[];              % ranks at rhp eigenvalues
rkb2=[];              % ranks at rhp eigenvalue

for j=1:p
rkb1=[rkb1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP1])];
rkb2=[rkb2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP2])];
end

if min(rkb1) < length(AP)
  sprintf('PROBLEM: (AP,BP1) IS NOT STABILIZABLE')
  stop
end

if min(rkb2) < length(AP)
  sprintf('PROBLEM: (AP,BP2) IS NOT STABILIZABLE')
  stop
end

% Detectability of (CP2,AP) and (CP1,AP)
% PHB test: [(-AP+sI)' CP2']' must not lose rank at closed RHP eigvals
of AP

rkc1=[];               % ranks at rhp eigenvalues
rkc2=[];               % ranks at rhp eigenvalues

for j=1:p
   rkc1=[rkc1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP1])];
   rkc2=[rkc2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP2])];
end

if min(rkc1) < length(AP)
  sprintf('PROBLEM: (CP1,AP) IS NOT DETECTABLE')
  stop
end

if min(rkc2) < length(AP)
  sprintf('PROBLEM: (CP2,AP) IS NOT DETECTABLE')
  stop
end

% There are two more conditions that should be checked. The matrices
% [AP-jwI BP2; CP1 DP12] and [AP-jwI BP1; CP2 DP21] must have full
column
% and row rank respectively for all w.

ww  = logspace(-5,2,200);

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP2; CP1 DP12]);
  if rk < (length(AP)+length(BP2(1,:)))
     sprintf('PROBLEM: FIRST MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end
```

```
for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP1; CP2 DP21]);
  if rk < (length(AP)+length(CP2(:,1)))
     sprintf('PROBLEM: SECOND MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end


% compute the H-infinity optimal controller
ricmethod=2;
tol=1e-3;
[K,Tzw,norms,X,Y,hamx,hamy]=hinfsyn(P,nmeas,ncon,0,2,tol,ricmethod);
```

muroomheating.m

```
%% (c) Coronado systems 2000
% This m-file was used to generate the mu plots of the room heating
% process with uncertainty in the sensor dynamics and the process time
constant
% the "small-mu" robustness condition is checked.

% frequency grid
ww=logspace(-5,1,200);

% nominal plant transfer function
numG=1;
denG=[1000 1];
G=nd2sys(numG,denG);
dcg=0.01;             % process DC gain

% nominal sensor transfer function
numGs=1;
denGs=[10 1];
Gs=nd2sys(numGs,denGs);

% weigthing function for mult. uncertainty
numWm=0.1;
denWm=1;
Wm=nd2sys(numWm,denWm);

% weigthing function for uncertainty in process time constant
numWf=[200 0];
denWf=[0.0001 1];
Wf=nd2sys(numWf,denWf);

% Controller (including negative sign)
numK=-1000;
denK=1;
K=nd2sys(numK,denK);


% Build the system interconnection to obtain the generalized plant P(s)
% using the "sysic" command
systemnames=' Wf Wm G Gs dcg';
inputvar='[ wf; wm; u ]';
outputvar='[ G; Gs; Wm + Gs ]';
input_to_Wf='[ wf ]';
input_to_Wm='[ wm ]';
input_to_G='[ dcg - Wf ]';
input_to_Gs='[ G ]';
input_to_dcg='[ u ]';
sysoutname='P';
cleanupsysic='yes';
sysic;

% Compute the FR of the LFT FL(P,K)
FLPK=starp(P,K);
FLPKjw=frsp(FLPK,ww);
```

```
% Define uncertainty block structure and compute bounds on mu
Gamma=[1 1; 1 1];     % two separate scalar (1x1) blocks
mubounds=mu(FLPKjw,Gamma);

% Plot upper and lower bounds on mu and check if it is lower than 1
figure(1)
vplot('liv,m',mubounds);

% Define uncertainty block structure for mixed-mu
% with a real perturbation of the time constant
Gamma1=[-1 0; 1 1];     % two separate scalar (1x1) blocks, the first
is real
mixedmubounds=mu(FLPKjw,Gamma1);

% Plot upper and lower bounds on mu and check if it is lower than 1
figure(2)
vplot('liv,m',mixedmubounds,mubounds,'-');


% Build a new system interconnection to obtain the generalized plant
P1(s)
% using the "sysic" command for robust performance analysis
K=mmult(-1,K);

% weigthing function for performance
numWe=[10];
denWe=[2500 1];
We=nd2sys(numWe,denWe);


systemnames=' Wf Wm We G Gs dcg';
inputvar='[ wf; wm; Tdes; u ]';
outputvar='[ G; Gs; We; Tdes - Wm - Gs ]';
input_to_Wf='[ wf ]';
input_to_Wm='[ wm ]';
input_to_We='[ Tdes - Wm - Gs ]';
input_to_G='[ dcg - Wf ]';
input_to_Gs='[ G ]';
input_to_dcg='[ u ]';
sysoutname='P1';
cleanupsysic='yes';
sysic;

% Compute the FR of the LFT FL(P1,K)
FLP1K=starp(P1,K);
FLP1Kjw=frsp(FLP1K,ww);

% Define uncertainty block structure and compute bounds on mu
Gamma2=[1 1; 1 1; 1 1];     % two separate scalar unc. blocks + 1
fictitious scalar perf. block
muperfbounds=mu(FLP1Kjw,Gamma2);

% Plot upper and lower bounds on mu and check if it is lower than 1
figure(3)
vplot('liv,m',muperfbounds);
```

musynroomheating.m

```
%% (c) Coronado systems 2000

% This m-file was to perform a mu synthesis for the room heating
% process with uncertainty in the sensor dynamics and the process time
constant
% the "small-mu" robustness condition is checked at the end.

% frequency grid
ww=logspace(-5,1,200);

% nominal plant transfer function
numG=1;
denG=[1000 1];
G=nd2sys(numG,denG);
dcg=0.01;             % process DC gain

% nominal sensor transfer function
numGs=1;
denGs=[10 1];
Gs=nd2sys(numGs,denGs);

% weigthing function for mult. uncertainty
numWm=0.1;
denWm=1;
Wm=nd2sys(numWm,denWm);

% weigthing function for uncertainty in process time constant
numWf=[200 0];
denWf=[0.0001 1];
Wf=nd2sys(numWf,denWf);

% weigthing function for performance
numWe=[10];
denWe=[2500 1];
We=nd2sys(numWe,denWe);

% biproper weighting function on control signal (almost always
required)
numWu=0.00001;
denWu=1;
Wu=nd2sys(numWu,denWu);


% Build the system interconnection to obtain the generalized plant P(s)
% using the "sysic" command for robust performance analysis


systemnames=' Wf Wm We Wu G Gs dcg';
inputvar='[ wf; wm; Tdes; u ]';
outputvar='[ G; Gs; We; Wu; Tdes - Wm - Gs ]';
input_to_Wf='[ wf ]';
```

```
input_to_Wm='[ wm ]';
input_to_We='[ Tdes - Wm - Gs ]';
input_to_Wu='[ u ]';
input_to_G='[ dcg - Wf ]';
input_to_Gs='[ G ]';
input_to_dcg='[ u ]';
sysoutname='P';
cleanupsysic='yes';
sysic;

% get a minimal balanced realization of P
[P,sig]=sysbal(P,1e-10)

% Extract partitioned state-space matrices
[AP,BP,CP,DP]=unpck(P);
BP1=BP(:,1:3);
BP2=BP(:,4);
CP1=CP(1:4,:);
CP2=CP(5,:);
DP11=DP(1:4,1:3);
DP12=DP(1:4,4);
DP21=DP(5,1:3);
DP22=DP(5,4);

% Stabilizability of (AP,BP2) and (AP,BP1)
% PBH test: [-AP+sI BPi] must not lose rank at closed RHP eigvals of AP
rhpeigs=[];
[V,EIGS] = eig(AP);
eigs = diag(EIGS);

for i=1:length(AP)
if real(eigs(i)) >= 0
      rhpeigs=[rhpeigs;eigs(i)];
end
end

p = length(rhpeigs);
r = length(AP);      % Rank of [sI-AP] for s not eigval(AP)
rkb1=[];             % ranks at rhp eigenvalues
rkb2=[];             % ranks at rhp eigenvalue

for j=1:p
rkb1=[rkb1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP1])];
rkb2=[rkb2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP BP2])];
end

if min(rkb1) < length(AP)
  sprintf('PROBLEM: (AP,BP1) IS NOT STABILIZABLE')
  stop
end

if min(rkb2) < length(AP)
  sprintf('PROBLEM: (AP,BP2) IS NOT STABILIZABLE')
  stop
end
```

```
% Detectability of (CP2,AP) and (CP1,AP)
% PHB test: [(-AP+sI)' CP2']' must not lose rank at closed RHP eigvals
of AP

rkc1=[];              % ranks at rhp eigenvalues
rkc2=[];              % ranks at rhp eigenvalues

for j=1:p
   rkc1=[rkc1 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP1])];
   rkc2=[rkc2 ; rank([(rhpeigs(j)*eye(length(AP)))-AP; CP2])];
end

if min(rkc1) < length(AP)
  sprintf('PROBLEM: (CP1,AP) IS NOT DETECTABLE')
  stop
end

if min(rkc2) < length(AP)
  sprintf('PROBLEM: (CP2,AP) IS NOT DETECTABLE')
  stop
end

% There are two more conditions that should be checked. The matrices
% [AP-jwI BP2; CP1 DP12] and [AP-jwI BP1; CP2 DP21] must have full
column
% and row rank respectively for all w.

ww  = logspace(-5,2,200);

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP2; CP1 DP12]);
  if rk < (length(AP)+length(BP2(1,:)))
     sprintf('PROBLEM: FIRST MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end

for i=1:200
  rk = rank([sqrt(-1)*ww(i)*eye(length(AP))-AP BP1; CP2 DP21]);
  if rk < (length(AP)+length(CP2(:,1)))
     sprintf('PROBLEM: SECOND MATRIX IS RANK DEFICIENT AT FREQ. =
%g',ww(i))
     stop
  end
end



% The uncertainty block structure is defined in the separate file
% musynroomdefn.m

% compute the H-infinity optimal controller as the starting point for
D-K iteration
nmeas=1; % number of measurements to controller
ncon=1;  % number of control signals from controller
```

```
ricmethod=2;
tol=1e-10;
[K,Tzw,norms,X,Y,hamx,hamy]=hinfsyn(P,nmeas,ncon,0,5,tol,ricmethod);

%lancer la dk iteration
DK_DEF_NAME='musynroomdefn';
dkit


% Compute the FR of the LFT FL(P1,K)
FLPK=starp(P,K);
FLPKjw=frsp(FLPK,ww);

% Define uncertainty block structure and compute bounds on mu
Gamma=[1 1; 1 1; 1 2];     % two separate scalar unc. blocks + 1
fictitious scalar perf. block
muperfbounds=mu(FLPKjw,Gamma);

% Plot upper and lower bounds on mu and check if it is lower than 1
figure(3)
vplot('liv,m',muperfbounds);
```

musynroomdefn.m

```
%-----------------------------------------------%
%     REQUIRED USER DEFINED VARIABLES        %
%-----------------------------------------------%
% Nominal plant interconnection structure
NOMINAL_DK = P;

% Number of measurements
NMEAS_DK = 1;

% Number of control inputs
NCONT_DK = 1;

% Block structure for mu calculation
BLK_DK = [1 1;1 1;1 2];

% Frequency response range
OMEGA_DK = logspace(-3,3,200);


%-------------------- end  --------------------------------%
```