20 LTI Feedback Design

In this lecture, the *standard H2 and H-Infinity output feedback design setup* is introduced in terms of of *finite order state space LTI models*, their *feedback interconnections*, *stability* and *system norms* (*H2 norm* and *H-Infinity norm*).

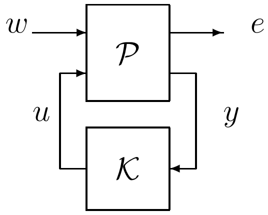


Figure 20.1: Standard feedback design diagram

The standard setup is frequently illustrated by the block diagram from Figure 20.1, where $\mathcal{P}$ represents the *plant*, a given finite order linear time invariant (LTI) state space system model with input partitioned into two vector signal components w (*disturbance*) and u *control*, and output partitioned into two vector signal components e (*cost*) and y (*sensor*). K is the *controller*, which is to be found, also in the format of a finite order LTI state space model, with input y and output u. The controller K must satisfy a condition ensuring *well-posedness* of the *interconnection* of P and K, in which case it defines the *closed loop system* G, also an LTI state space model, with input w and output e. The criteria for selecting K are formulated in terms of *stability* and *norm minimization* of the closed loop system in terms of either H-Infinity of H2 norm.

The canonical setup is a major component of modern feedback design approach: application problems involving complex models and advanced specifications are frequently solved by reduction to standard H2/H-Infinity optimization.

20.1 Interconnections of State Space Models

Systems can be imagined as boxes with input and output wires coming in and out. When several such boxes are available, some of the output wires can be connected to some of the available inputs. Under certain conditions, this results in an *interconnection* defining a new state space model, its state vector being concatenation of state vectors of the individual subsystems.

The *series interconnection* of two systems S_2 and S_1 is a new system (denoted by S_2S_1) which maps the input $f = f_1$ of S_1 into the output $y = y_2$ of S_2 under the assumption that the output y_1 of S_1 is identical to the input f_2 of S_2 (see Figure 20.2).

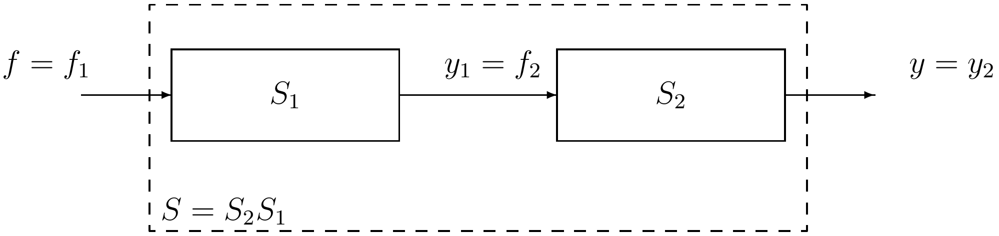


Figure 20.2: Series Interconnection of S_2 and S_1

A formal definition of a series connection of two state space LTI models is as follows.

Definition 20.1 A *series interconnection* $S = S_2 S_1$ of state space models

$$S_2 = \mathcal{S}[a_2, b_2, c_2, d_2], \quad S_1 = \mathcal{S}[a_1, b_1, c_1, d_1],$$

where the domains of S_1, S_2 are the same (either both DT or both CT), and the number of columns of matrix d_2 equals the number of rows of d_1, is state space model $S = \mathcal{S}[a, b, c, d]$ with the same time domain as S_1 and S_2, and with matrices

$$a = \begin{bmatrix} a_2 & b_2 c_1 \\ 0 & a_1 \end{bmatrix}, \quad b = \begin{bmatrix} b_2 d_1 \\ b_1 \end{bmatrix}, \quad c = \begin{bmatrix} c_2 & d_2 c_1 \end{bmatrix}, \quad d = d_2 d_1.$$

For example, the interconnection of two continuous time state space models

$$S_2 = \mathcal{S}_{CT}[a_2, b_2, c_2, d_2],$$

with input f_2, output y_2, and state x_2, and $S_1 = \mathcal{S}_{CT}[a_1, b_1, c_1, d_1]$, with input f_1, output y_1, and state x_1, define continuous time LTI state space equations with input $f = f_1$, output $y = y_2$ and state $x = [x_2; x_1]$ which come as the result of replacing f_2 with $y_1 = c_1 x_1 + d_1 f_1$ in

$$\dot{x}_1 = a_1 x_1 + b_1 f_1, \quad y_1 = c_1 x_1 + d_1 f_1,$$

$$\dot{x}_2 = a_2 x_2 + b_2 f_2, \quad y_2 = c_2 x_2 + d_2 f_2.$$

Taking into account the interpretation of transfer matrices, transfer matrix of a series interconnection is the product of transfer matrices of the subsystems.

Theorem 20.1 *If $S = S_2 S_1$ is a series interconnection of state space LTI models then $G = G_2 G_1$, where G, G_1, G_2 are transfer matrices of S, S_1, S_2 respectively.*

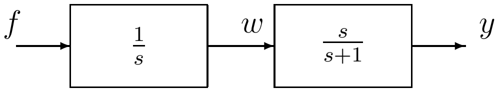


Figure 20.3: Series interconnection of S_a and S_b

Example 20.1 LTI CT state space model $S_a = \mathcal{S}[-1,1,-1,1]$ has transfer function $G_a(s) = s/(s+1)$. Since G_a has a *zero* at $s=0$, i.e. $G_a(0)=0$, it is practically impossible to produce a constant non-zero output while applying a bounded input to S_a. It could be tempting to alter the situation by using a series interconnection $S_{ab} = S_aS_b$ or $S_{ba} = S_bS_a$ of S_a with system $S_b = \mathcal{S}[0,1,1,0]$. Indeed, S_b has transfer function $G_b(s) = 1/s$, and hence both S_{ab} and S_{ba} have transfer function $\frac{1}{s+1}$, which is stable and has no zero at $s=0$.

Detailed analysis of the situation reveals a less attractive picture. For the series interconnection of S_a and S_b (see Figure 20.3) we have $S_aS_b = \mathcal{S}[A,B,C,D]$, where

$$A = \begin{bmatrix} -1 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} -1 & 1 \end{bmatrix}, \quad D = 0.$$

This state space model is not minimal, because the pair (C,A) is not observable. Moreover, the unobservable part of system dynamics is unstable, which, essentially, predicts that bounded input noise will drive the state w of the first stage S_b of the series interconnection unbounded.

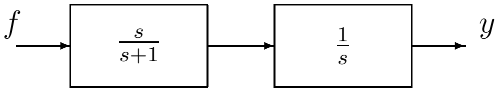


Figure 20.4: Series interconnection of S_b and S_a

A similar outcome is observed in the series interconnection of S_b and S_a (see Figure 20.4) we have $S_aS_b = \mathcal{S}[A,B,C,D]$, where

$$A = \begin{bmatrix} 0 & -1 \\ 0 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad D = 0.$$

This state space model is not minimal, because the pair (A,B) is not controllable. Moreover, the uncontrollable part of system dynamics is unstable, which predicts that, even with a zero input f, certain initial conditions will drive output y unbounded.

The *feedback interconnection* of systems S_1 and S_2 (see Figure 20.5) can be considered in the case when the dimension of input f_1 of S_1 is larger than dimension of output y_2 of S_2, and the dimension of output y_1 of S_1 is larger than dimension of output y_2 of S_2. Then f_1, y_1 can be partitioned according to

$$f_1 = \begin{bmatrix} f_{11} \\ f_{12} \end{bmatrix}, \quad y_1 = \begin{bmatrix} y_{11} \\ y_{12} \end{bmatrix},$$

where signals f_{12} and y_2 have same dimension, and the dimension of y_{12} equals the dimension of f_2 as well. The interconnection is expected to define a system with input $f = f_{11}$ and output $y = y_{11}$, for which input f produces output y if and only if there exist signals f_2, y_2 such that y_2 is a response of S_2 to input f_2, and $[y; f_2]$ is a response of S_1 to input $[f; y_2]$.

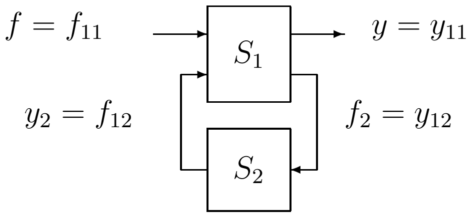


Figure 20.5: Feedback interconnection of S_1 and S_2

Example 20.2 Let S_1 be the system with input $f_1 = [f_{11}; f_{12}]$ and output $y_1 = [y_{11}; y_{12}]$ (where f_{11}, f_{12}, y_{11}, y_{12} are scalar signals) defined by

$$y_{11} = f_{11} + f_{12}, \quad y_{12} = f_{11} + f_{12}. \tag{20.1}$$

Let S_2 be the system with scalar input y_{12} and scalar output f_{12}, defined by

$$f_{12} = k y_{12}, \tag{20.2}$$

where k is a real parameter.

When $k \neq 1$, the feedback interconnection of S_1 and S_2 is the memoryless LTI system with input f_{11} and output y_{11} defined by

$$y_{11} = \frac{2-k}{1-k} f_{11}.$$

When $k = 1$, the feedback interconnection does not define an input/output transformation $f_{11} \mapsto y_{11}$, because equations (20.1) and (20.2) together impose a *constraint* $f_{11} \equiv 0$ on the input signal. Accordingly, a feedback interconnection of two LTI state space models does not always define a state space model.

In MATLAB, equations for interconnections of finite order LTI models are can be produced using function `lft.m`. For example,

```
lft(ss([],[],[],[1 1;1 1]),ss([],[],[],0))
```

produces the identity transformation $\mathcal{S}_{CT}[\emptyset, \emptyset, \emptyset, 1]$, while

```
lft(ss([],[],[],[1 1;1 1]),ss([],[],[],1))
```

yields an error message.

Definition 20.2 A *feedback interconnection* of state space models $S_1 = \mathcal{S}[a_1, b_1, c_1, d_1]$ and $S_2 = \mathcal{S}[a_2, b_2, c_2, d_2]$, for which the domains of S_1, S_2 are the same (either both DT or both CT), the number of columns of matrix d_1 is larger than the number m of rows of d_2, and the number of rows of d_1 is larger than the number k of columns of d_2, is called *well-posed* when matrix $I_k - d_{122}d_2$ is not singular, where $\tilde{d_1}$ is the lower right k-by-m corner submatrix of d_1, i.e.

$$d_1 = \begin{bmatrix} d_{111} & d_{112} \\ d_{121} & d_{122} \end{bmatrix}.$$

A well-posed feedback interconnection of S_1 and S_2 defines a state space model S with state vector $x = [x_1; x_2]$, where x_k for $k \in \{1, 2\}$ enotes the state of S_k.

20.2 H2 and H-Infinity Norms

H2 and H-Infinity norms quantify sensitivity of LTI system output to its input. H-Infinity norm of a transfer matrix G is the smallest upper bound for the gain of $G(s)$ over all s with positive real part. According to this definition, H-Infinity norm is infinite for a rational transfer matrix with a pole in the closed right half plane. For a rational transfer matrix with no poles in the closed right half plane, H-Infinity norm equals the *L-Infinity norm*, defined as the smallest upper bound for the gain of $G(s)$ over the imaginary axis. The importance of H-Infinity norm as a sensitivity measure stems from the fact that it equals L2 gain of the corresponding LTI system.

H2 norm of a transfer matrix G is formally defined by

$$\|G\|_{H2}^2 = \sup_{h>0} \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}[G(j\omega + h)' G(j\omega + h)] d\omega.$$

For a rational transfer matrix, this yields $+\infty$ when there is a pole in the closed right half plane, and the *L2 norm* $\|G\|_{L2}$ of G, defined by

$$\|G\|_{L2}^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \text{trace}[G(j\omega)' G(j\omega)] d\omega$$

otherwise. H2 norm of a stable transfer matrix G can be interpreted as the limit value of $\mathbf{E}[|y(t)|^2]$ as $t \to +\infty$, where $y = y(t)$ is the response of the LTI system defined by G to a "normalized white noise" input f.

Using the Parceval formula, H2 norm of a rational strictly proper transfer matrix G given by its state space model coefficients can be computed exactly.

Theorem 20.2 *Let A, B, C be real matrices of dimensions n-by-n, n-by-m, and k-by-n respectively. Assume that A is a Hurwitz matrix. Then*

$$\|G\|_{H2}^2 = \text{trace}[B' W_o B] \ \text{ for } \ G(s) = C(sI - A)^{-1} B, \ \ W_o A + A' W_o = -C'C.$$

Proof. For every k-by-m complex matrix M

$$\mathrm{trace}(M'M) = \sum_{i=1}^{k} |Me_i|^2$$

where $\{e_i\}_{i=1}^{k}$ is the standard basis in $\mathbb{R}^m$. Hence, using the Parceval formula,

$$\|G\|_{H2}^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} |G(j\omega)e_i|^2 d\omega = \int_0^{\infty} |Cx_i(t)|^2 dt,$$

where $x_i = x_i(t)$ are the solutions of

$$\dot{x}_i(t) = Ax_i(t), \quad x_i(0) = Be_i.$$

Therefore

$$\|G\|_{H2}^2 = \sum_{i=1}^{m} e_i' B' W_c B e_i = \mathrm{trace}[B'W_oB],$$

where W_o is the observability Gramian of the pair (C, A). ■

Practical calculation of the H-Infinity norm is a bit trickier.

Theorem 20.3 *Assume that*

$$G(s) = D + C(sI - A)^{-1}B,$$

where A, B, C, D are real matrices of dimensions n-by-n, n-by-m, k-by-n, and k-by-m respectively, A has no eigenvalues on the imaginary axis, and $\gamma > 0$. The following conditions are equivalent:

(a) L-Infinity norm of G is less than γ;

(b) matrix $R \overset{\mathrm{def}}{=} \gamma^2 I - D'D$ is positive definite, and matrix

$$H = \begin{bmatrix} A + BR^{-1}D'C & -BR^{-1}B' \\ C'C + C'DR^{-1}D'C & -A - C'DR^{-1}B' \end{bmatrix}$$

has no eigenvalues on the imaginary axis.

Proof. Since $G(s) \to D$ as $s \to \infty$, condition (a) implies positive definiteness of R. On the other hand, when $R > 0$, L-Infinity norm of G is less than γ if and only if the matrix $\gamma^2 I - G(j\omega)'G(j\omega)$ is not singular for all $\omega \in \mathbb{R}$. Therefore, to finish the proof it is sufficient to show that $\gamma^2 I - G(j\omega)'G(j\omega)$ is not singular for all $\omega \in \mathbb{R}$ if and only if H has no eigenvales on the imaginary axis.

Indeed, the condition

$$\gamma^2 f = G(j\omega)'G(j\omega)f$$

is equivalent to

$$\gamma^2 f = (D' - B'(j\omega I + A')^{-1}C')(D + C(j\omega I - A)^{-1}B)f,$$

and hence can be re-written in the form

$$D'v - B'x_2 = \gamma^2 f,$$

where

$$v = (D + C(j\omega I - A)^{-1}B)f = Df + Cx_1, \quad x_1 = (j\omega I - A)^{-1}Bf, \quad x_2 = (j\omega I + A')^{-1}C'v,$$

i.e.

$$j\omega x_1 = Ax_1 + Bf, \quad j\omega x_2 = -A'x_2 + C'v, \quad v = Cx_1 + Df.$$

Combined together, these equations yield an equivalent equality

$$H \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = j\omega \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

which proves the theorem. ■

Example 20.3 L2 norm of $G(s) = 1/(s-a)$ equals $|2a|^{-1/2}$. H2 norm of $G(s) = 1/(s-a)$ equals $|2a|^{-1/2}$ for $a < 0$ and plus infinity otherwise. L-Infinity norm of $G(s) = 1/(s-a)$ equals $|a|^{-1}$ (the maximum of amplitude response is acieved at $\omega = 0$). L2 norm of $H(s) = s/(s+1)$ equals plus infinity (since $H(j\omega)$ does not converge to zero as $\omega \to \infty$). L-Infinity norm of $H(s) = s/(s+1)$ equals 1 (the maximum of amplitude response is acieved at $\omega = \infty$). H2 norm of $1 + 1/z$ equals $\sqrt{2}$, while L-Infinity norm of $1 + 1/z$ equals 2. In MATLAB, the calculations can be done according to

```
s=tf('s'); z=tf('z');
norm(1/(s-1))      % CT H2 norm
norm(1/(s+1))      % CT H2 norm
norm(1/(s-1),Inf) % CT L-Infinity norm
norm(s/(s+1))      % CT H2 norm
norm(s/(s+1),Inf) % CT L-Infinity norm
norm(1+1/z)        % DT H2 norm
norm(1+1/z,Inf)    % DT L-Infinity norm
```

20.3 Standard H2 Optimization

The standard task of H2 optimization is formulated in the following way: given a state space model of system P (see Figure 20.1), called *plant*, find a state space LTI model K called *controller* such that the the feedback interconnection of P and K is well-posed,

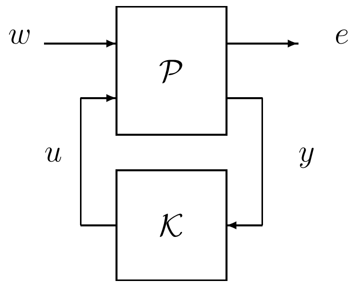


Figure 20.6: Standard feedback design diagram

defines a stable state space model G, and, under these assumptions, minimizes the H2 norm of G.

Let the plant equations have the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_1 w(t) + B_2 u(t), && (20.3) \\ e(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t), && (20.4) \\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t). && (20.5) \end{aligned}$$

Practically, the MATLAB's algorithm of H2 optimization implemented in `h2syn.m` applies only in the case when several additional conditions are satisfied:

(CT1) the pair (A, B_2) is stabilizable, i.e has no uncontrollable modes correspondig to the eigenvalues of A with non-negative real part;

(CT2) the pair (C_2, A) is detectable, i.e has no unobservable modes correspondig to the eigenvalues of A with non-negative real part;

(CT3) matrix D_{12} is left invertible;

(CT4) matrix

$$E_c(s) = \begin{bmatrix} A - sI & B_2 \\ C_1 & D_{12} \end{bmatrix}$$

is left invertible for all $s = j\omega$, $\omega \in \mathbb{R}$;

(CT5) matrix D_{21} is right invertible;

(CT6) matrix

$$E_m(s) = \begin{bmatrix} A - sI & B_1 \\ C_2 & D_{21} \end{bmatrix}$$

is right invertible for $s = j\omega$ for all $\omega \in \mathbb{R}$;

(CT7) $D_{22} = 0$;

(CT8) $D_{11} = 0$;

Conditions (CT1) and (CT2) are necessary for existence of a stabilizing controller K. Constraints (CT3)-(CT6), which can be referred to as conditions of *well-posedness* or *non-singularity* of the optimization task, are used (though they are not always necessary) to guarantee *existence of optimal controller* K. Condition (CT7) is very non-restrictive, and is needed to guarantee existence of optimal controller, though it is far from necessary. The need for condition (CT8) is another weakness of the algorithm: it is used to guarantee existence of a controller which makes closed loop H2 norm finite, but is far from necessary for that purpose. When conditions (CT1)-(CT8) are satisfied, function `h2syn.m` is supposed to return the optimal controller K, which always has same order as P. Practically, due to numerical errors, for a poorly conditioned setting `h2syn.m` can return an incorrect answer.

For the discrete time case, when plant equations have the form

$$\begin{aligned} x(t+1) &= Ax(t) + B_1 w(t) + B_2 u(t), && (20.6)\\ e(t) &= C_1 x(t) + D_{11} w(t) + D_{12} u(t), && (20.7)\\ y(t) &= C_2 x(t) + D_{21} w(t) + D_{22} u(t). && (20.8) \end{aligned}$$

application of `h2syn.m` will require the following conditions:

(DT1) the pair (A, B_2) is stabilizable, i.e has no uncontrollable modes z with $|z| \geq 1$;

(DT2) the pair (C_2, A) is detectable, i.e has no unobservable modes z with $|z| \geq 1$;

(DT3) matrix D_{12} is left invertible;

(DT4) matrix

$$E_c(z) = \begin{bmatrix} A - zI & B_2 \\ C_1 & D_{12} \end{bmatrix}$$

is left invertible for all $z = e^{j\omega}$, $\omega \in [0, \pi]$;

(DT5) matrix D_{21} is right invertible;

(DT6) matrix

$$E_m(z) = \begin{bmatrix} A - zI & B_1 \\ C_2 & D_{21} \end{bmatrix}$$

is right invertible for $z = e^{j\omega}$ for all $\omega \in [0, \pi]$;

(DT7) $D_{22} = 0$;

(DT8) $D_{11} = 0$;

Conditions (DT1)-(DT8) have the same meaning as their CT counterparts (CT1)-(CT8), with the only exception that (DT8) becomes absolutely unnecessary.

20.4 Standard H-Infinity Optimization

The standard task of H-Infinity optimization is formulated in the following way: given a plant model P (see Figure 20.6), find controller model K satisfying the following conditions:

(Inf1) feedback interconnection of P and K is well-posed;

(Inf2) the closed loop state space model G is stable;

(Inf3) H-Infinity norm of G is smaller than $\gamma = (1+\epsilon)\gamma_{\min}$, where $\gamma_{\min}$ is the maximal lower bound for the H-Infinity norm of G achievable subject to constraints (Inf1), (Inf2), and $\epsilon > 0$ is a pre-specified tolerance.

Practically, the MATLAB's algorithm of H-Infinity optimization implemented in function `hinfsyn.m` applies only when conditions (CT1) - (CT7) are satisfied in the CT case, and conditions (DT1) - (DT7) are satisfied in the DT case. Just as `h2syn.m` (even to a higher degree), `hinfsyn.m` can return an incorrect answer for a poorly conditioned setting.

Example 20.4 Consider the task of minimizing H-Infinity norm of the closed loop system for the plant

$$e(t) = y(t) = u(t) + w(t), \tag{20.9}$$

transfer matrix

$$P(s) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Using feedback $u(t) = D_f y(t)$ with $D_f \neq 1$ yields $e(t) = y(t) = (1-D_f)^{-1}w(t)$. Hence H-Infinity norm of the closed loop system can be made arbitrarily small when $D_f \to \infty$. However, the zero lower bound is not achieved by any particular admissible controller.

20.4.1 Example: non-unique H-Infinity optimal controller

Consider the H-Infinity feedback optimization setup with two-dimensional signals w, u, e, y, defined by the plant transfer matrix

$$P(s) = \begin{bmatrix} \frac{2}{s+1} & 0 & \frac{s-1}{s+1} & 0 \\ 0 & \frac{1}{s+1} & 0 & \frac{s-1}{s+1} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

Since $P_{22} = 0$ is this example, there is no actual feedback loop to close, and hence a feedback controller $K = K(s)$ is stabilizing if and only if it is stable. Accordingly, the closed loop transfer matrix G is given by

$$G(s) = \begin{bmatrix} \frac{2}{s+1} & 0 \\ 0 & \frac{1}{s+1} \end{bmatrix} + \begin{bmatrix} \frac{s-1}{s+1} & 0 \\ 0 & \frac{s-1}{s+1} \end{bmatrix} K(s).$$

Substituting $s = 1$ into the expression for G yields

$$G(1) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

Hence $\|G\|_\infty \geq \|G(1)\| \geq 1$ for all stabilizing controllers K. On the other hand, using

$$K(s) = K_0(s) = \begin{bmatrix} 1 & 0 \\ 0 & 1/2 \end{bmatrix}$$

produces

$$G(s) = G_0(s) = \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

Since $\|G_0\|_\infty = 1$, K_0 is an H-Infinity optimal controller in this setup. However, K_0 is not the only H-Infinity optimal controller: for example, it is easy to check that

$$K_1(s) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

is optimal as well.

20.4.2 Example: L2 gain optimization via small gain theorem

Consider the task of designing a feedback controller for the standard setup shown on Figure 20.7, where H is an unstable LTI plant with delay,

$$H(s) = \frac{e^{-\tau s}}{s-1},$$

and K is the controller to be designed to guarantee good tracking of the reference input r at lower frequencies.

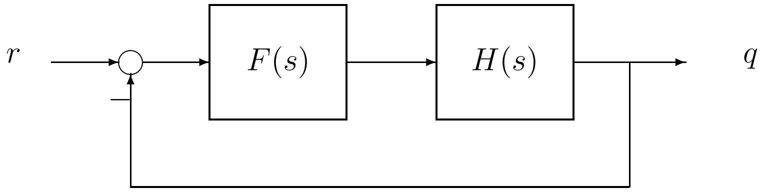


Figure 20.7: Feedback design with delay

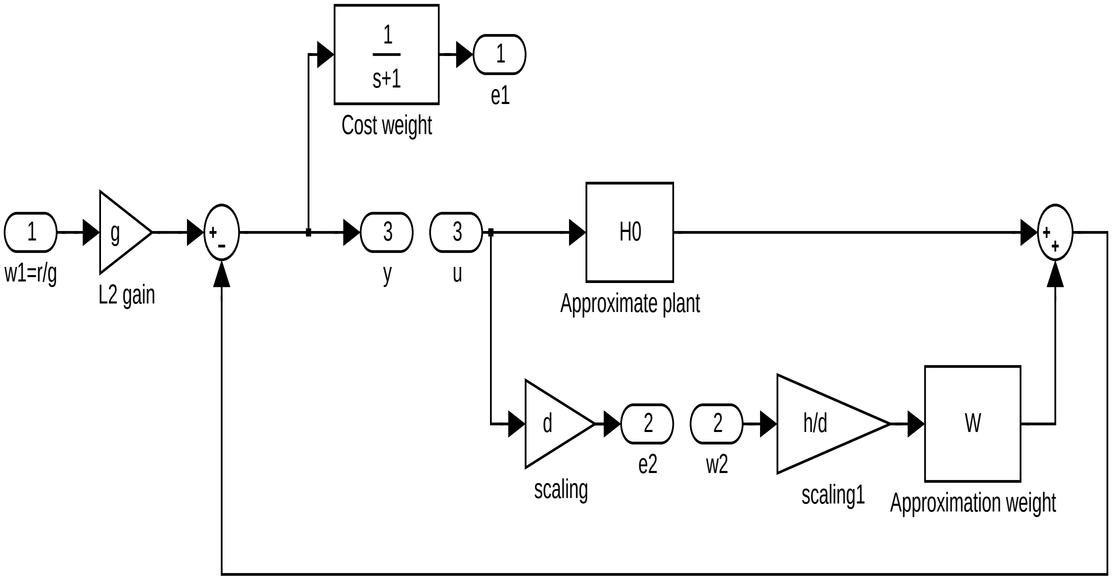
We begin by approximating the stable part of H by a lower order transfer function, while bounding the approximation error:

$$H(s) = H_0(s) + hW(s)\Delta(s),$$

where

$$H_0(s) = \frac{e^{-\tau}}{s-1} - \frac{\tau}{1+0.5\tau}\frac{1}{1+0.5\tau s},$$
$$W(s) = \frac{1+s}{10+s}, \quad h = \|(H-H_0)/W\|_\infty,$$

and $\Delta(s)$ is known to have H-Infinity norm not larger than one. The corresponding feedback design diagram and MATLAB code are shown below.



Here γ is the desired guaranteed closed loop L2 gain from the reference input $r = \gamma w_1$ to the frequency weighted tracking error e_1. Note the way in which the additional scaling parameter d is used. H-Infinity optimization does not provide for an efficient way of selecting the weights, including, in this case, d and γ. This particular implementation relies on manual adjustment of d and γ.

```
function [K,G]=b05_ex7(tau,g,d)
% function [K,G]=b05_ex7(tau,g,d)
%
% L2 gain optimization via small gain theorem
if nargin<1, tau=0.1; end      % default delay value
if nargin<2, g=5; end          % default target L2 gain
if nargin<3, d=1/150; end      % default scaling parameter
s=tf('s');       % useful "transfer function"
H0=exp(-tau)/(s-1)-(tau/(1+tau/2))/(1+tau*s/2);  % approximation of H
W=(1+s)/(1+s/10);                                % frequency weight
```

```
w=linspace(0,100,10000);                         % frequency samples
H0w=squeeze(freqresp(H0,w)).'; % frequency response of H0
Ww=squeeze(freqresp(W,w)).';  % frequency response of W0
Hw=exp(-tau*j*w)./(j*w-1);     % frequency response of H
h=max(abs((H0w-Hw)./Ww));      % approximate weighted H-Infinity error
disp(['Weighted H-Infinity error: ' num2str(h)])
assignin('base','H0',H0);      % workspace assignments
assignin('base','W',W);
assignin('base','g',g);
assignin('base','d',d);
assignin('base','h',h);
P=linmod('b05_ex7mod');   % extract plant model from SIMULINK diagram
p=pck(P.a,P.b,P.c,P.d);   % plant model in Mutools format
nmeas=1;                  % dimension of y
ncon=1;                   % dimension of u
gmin=0;                   % lower bound of the binary search interval
gmax=1.1;                 % upper bound of the binary search interval
tol=0.01;                 % relative accuracy of binary search
[k,g]=hinfsyn(p,nmeas,ncon,gmin,gmax,tol);   % controller optimization
[ak,bk,ck,dk]=unpck(k);           % conversion from Mutools format
[ag,bg,cg,dg]=unpck(g);
K=ss(ak,bk,ck,dk);
G=ss(ag,bg,cg,dg);
disp(['Closed loop L2 gain: ' num2str(norm(G,Inf))])
```