Chapter 5

Standard LTI Feedback Optimization Setup

Efficient LTI feedback optimization algorithms comprise a major component of modern feedback design approach: application problems involving complex models and advanced specifications are frequently solved by reduction to H2 or H-Infinity optimization. This chapter discusses the format and challenges of setting up a canonical LTI optimization task.

5.1 The Canonical Setup

The standard LTI feedback optimization setup can be described by the block diagram from Figure 5.1. In this diagram, block P represents the *plant*, an LTI model defined by

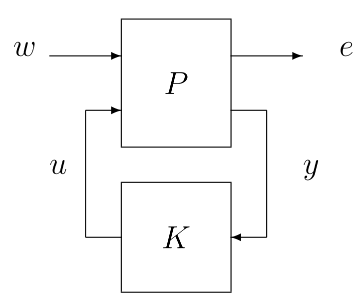

Figure 5.1: Standard feedback design diagram

a finite dimensional state space model

$$x^+(t) = Ax(t) + B_1 w(t) + B_2 u(t), \tag{5.1}$$
$$e(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t), \tag{5.2}$$
$$y(t) = C_2 x(t) + D_{21} w(t), \tag{5.3}$$

where $x^+(t)$ stands for $x(t+1)$ in the discrete time case, or $\dot{x}(t)$ in the continuous time case, and the coefficient matrices A, B_i, C_i, D_{ik} are known. In turn, block K represents the *controller*, an LTI model defined by a finite dimensional state space model

$$x_f^+(t) = A_f x_f(t) + B_f y(t), \tag{5.4}$$
$$u(t) = C_f x_f(t) + D_f y(t), \tag{5.5}$$

where the coefficients A_f, B_f, C_f, D_f are to be designed.

The input of P is partitioned into two vector signal components w (*disturbance*) and u (*control*, also the total output of K). The output of P is partitioned into two vector signal components e (*cost*) and y (*sensor*, also the input of K).

Since $u(t)$ does not enter directly into the expression for $y(t)$ in (5.3), the block diagram defines a valid closed loop state space model

$$x_{cl}^+(t) = A_{cl} x_{cl}(t) + B_{cl} w(t), \tag{5.6}$$
$$e(t) = C_{cl} x_{cl}(t) + D_{cl} w(t), \tag{5.7}$$

where

$$x_{cl} = \begin{bmatrix} x \\ x_f \end{bmatrix}, \quad B_{cl} = \begin{bmatrix} B_1 + B_2 D_f D_{21} \\ D_f D_{21} \end{bmatrix}, \quad A_{cl} = \begin{bmatrix} A + B_2 D_f C_2 & B_2 C_f \\ B_f C_2 & A_f \end{bmatrix},$$

$$C_{cl} = \begin{bmatrix} C_1 + D_{12} D_f C_2 & D_{12} C_f \end{bmatrix}, \quad D_{cl} = D_{11} + D_{12} D_f D_{21}.$$

The controller K is required to be *stabilizing*, in the sense that A_{cl} is a Hurwitz matrix (in the CT case) or a Schur matrix (in the DT case).

Finally, a standard feedback design setup has an *objective* formulated as a real-valued function of the closed loop model (5.6),(5.7), to be minimized with respect to K, subject to the constraints of well-posedness and stabilization. The classical results of robiust control deal with *H2 optimization* (implemented in MATLAB's `h2syn.m`) and *H-Infinity optimization* (implemented in MATLAB's `hinfsyn.m`).

H2 optimization is the task of minimizing the H_2 norm of the closed loop transfer matrix

$$G(\lambda) = D_{cl} + C_{cl}(\lambda I - A_{cl})^{-1} B_{cl}. \tag{5.8}$$

Recall that *H2 norm* $\|G\|_{H2}$ of a transfer matrix G is given by

$$\|G\|_{H2}^2 \stackrel{\text{def}}{=} \frac{1}{2\pi}\int_{-\infty}^{\infty} \|G(e^{j\omega})\|_F^2 d\omega \tag{5.9}$$

in the CT case, where $\|M\|_F^2 = \text{trace}(M'M)$ is the Frobenius norm of matrix M (i.e. the sum of squares of absolute values of its elements), or by

$$\|G\|_{H2}^2 \stackrel{\text{def}}{=} \frac{1}{2\pi}\int_{-\pi}^{\pi} \|G(e^{j\Omega})\|_F^2 d\Omega \tag{5.10}$$

in the DT case.

H-Infinity optimization is the task of minimizing *H-Infinity* norm $\|G\|_\infty$ of the closed loop transfer matrix G:

$$\|G\|_\infty^2 \stackrel{\text{def}}{=} \max_{\lambda\in\Omega_0} \sigma_{\max}(G(\lambda)), \tag{5.11}$$

where $\overline{\Omega_0} = j\mathbb{R}\cup\infty$ in the CT case, and $\overline{\Omega_0} = \mathbb{T}$ in the DT case.

5.2 Well-Posedness of LTI Feedback Optimization

Not every set of coefficient matrices A, B_i, C_i, D_{ik} of the plant are suitable for feedback optimization. In this section we discuss the typical constraints to be imposed on the plant.

5.2.1 Stabilizability

Obviously, an LTI feebdack optimization setup does not make much sense when there is no stabilizing controller K, in which case the set to optimiza over is empty. The classical control theory establishes necessary and sufficient conditions for the existence of a stabilizing controller.

(SC) the pair (A, B_2) is *stabilizable*, i.e. there exists matrix F such that $A+B_2F$ is either a Schur matrix (in the DT case) or a Hurwitz matrix (in the CT case);

(SO) the pair (C_2, A) is detectable, i.e. there exists matrix L such that $A+LC_2$ is either a Schur matrix (in the DT case) or a Hurwitz matrix (in the CT case).

5.2.2 Existence of an Argument of Minimum

Not all optimization tasks has a well defined *minimizer*, i.e. an argument of minimum. For example, the task of minimizing $\Phi(x) = \exp(x)$ over $x \in \mathbb{R}$ is ill posed, in the sense that the maximal lower bound of he set $\{\Phi(x) : \ x \in \mathbb{R}\}$, i.e. $\sup_{x \in \mathbb{R}} \Phi(x)$, which equals 0, is not achieved at any $x \in \mathbb{R}$. Instead, any sequence of real numbers $\{x_k\}_{k=1}^{\infty}$ such that $\Phi(x_k) \to 0$ must converge to $-\infty$.

A situation like this is relatively common when an LTI feedback optimization task is formulated without sufficient care: the maximal lower bound for H2 or H-Infinity norm of the closed loop system may not be achievable within the class of "good" controllers. This has to be avoided, either because the standard H2 or H-Infinity optimization algorithms tend to fail when applied to such settings, or because the resulting controllers have undesirable characteristics (zero/pole cancellation, sensitivity to round-off errors, etc.)

The following conditions guarantee existence of a minimizer, at least in H2 or H-Infinity optimization:

(MC) matrices $E_c(\lambda)$, where $E_c(\infty) = D_{12}$, and

$$E_c(\lambda) = \begin{bmatrix} A - \lambda I & B_2 \\ C_1 & D_{12} \end{bmatrix},$$

for $\lambda \in \mathbb{C}$, are left invertible for all $\lambda \in \overline{\Omega_0}$.

(MO) matrices $E_o(\lambda)$, where $E_o(\infty) = D_{21}$, and

$$E_o(\lambda) = \begin{bmatrix} A - \lambda I & B_1 \\ C_2 & D_{21} \end{bmatrix},$$

for $\lambda \in \mathbb{C}$, are right invertible for all $\lambda \in \overline{\Omega_0}$.

Informally speaking, condition (MC) means that the control effort is penalized (i.e. enters into the total cost with a non-zero weight) at every frequency (including $\omega = \infty$ in the CT case). For example, if A has no eigenvalues on $\overline{\Omega_0}$ then (MC) is equivalent to left invertibility of

$$P_{12}(\lambda) = D_{12} + C_1(\lambda I - A)^{-1} B_2$$

(i.e. the open loop transfer matrix from u to e) for $\lambda \in \overline{\Omega_0}$.

Similarly, condition (MO) means that the sensor measurement is noisy at every frequency. For example, if A has no eigenvalues on $\overline{\Omega_0}$ then (MO) is equivalent to right invertibility of

$$P_{21}(\lambda) = D_{21} + C_2(\lambda I - A)^{-1} B_1$$

(i.e. the open loop transfer matrix from w to y) for $\lambda \in \overline{\Omega_0}$.

We will refer to a failure of condition (MC) at a particular frequency (DT or CT) as *control singularity* of the feedback optimization setup at that frequency. Similarly, a failure of (MO) will be referenced as *observation singularity*. Examples describing simple basic cases of feedback optimization singularity are given in the next section.

5.2.3 Finiteness of H2 Norm

A peculiar thing about the H2 norm is that, in the CT case, it can easily equal $+\infty$ in a stabilized closed loop system: indeed, $\|G\|_\infty = \infty$ unless $D_{cl} = D_{11} - D_{12}D_fD_{21} = 0$. To simplify things, MATLAB's function `h2syn.m`, which implements H2 optimization, requires $D_{11} = 0$ (and produces the optimal controller with $D_f = 0$). Amazingly, it requires (at least in some of its versions) the same condition in the DT case, where it has no proper motivation at all.

5.3 Examples

The standard algorithm of H2 optimization (e.g. MATLAB's `h2syn.m`), requires a well-posed optimization setup (conditions (SC), (SO), (MC), (MO), $D_{11} = 0$), and produces the unique optimal controller (up to the accuracy of the numerical linear algebra operations involved). In contrast, the standard algorithm of H-Infinity optimization (e.g. MATLAB's `hinfsyn.m`), does not insist on a fully well-posed optimization setup (conditions (SC), (SO) are required, while some singularity is admissible, depending on the choice of `'method'`), but the resulting controller is not optimal, with the relative tolerance in reaching the maximal lower bound of $\|G\|_\infty$ being a parameter of the algorithm, which should not be made too small due to a certain danger of numerical instability. This section presents some useful examples of H2 and H-Infinity optimization, from both theoretical and numerical viewpoints.

5.3.1 Dangers of $D_{22} \neq 0$

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

$$e(t) = y(t) = u(t) + w(t), \tag{5.12}$$

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 controller to make the closed loop system well defined.

5.3.2 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 $P_{22} = 0$ in 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.

5.3.3 H2/H-Infinity Norms and Loop Shaping

According to their definition, H2 and H-Infinity norms show how *large* system's frequency response is across the frequency range: H2 norm applies a mean square *integral* measure, while H-Infinity norm is the maximal amplitude. Accordingly, H2 and H-Infinity optimization can be used to shape the closed loop transfer functions, forcing them to approximate a desired ideal response.

In their pure form, H2 or H-Infinity norm treat all frequencues the same. Using *weights* allows one to set preferences across frequencies.

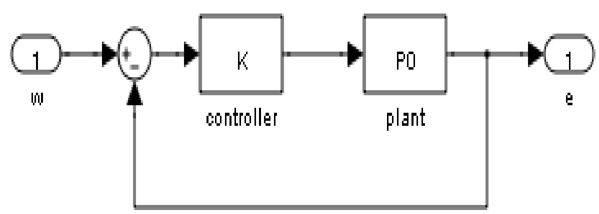

Figure 5.2: A simple feedback design setup

Consider, for example, the task of designing a stabilizing SISO feedback $K = K(s)$ to minimize reference tracking error and sensitivity to sensor noise in the block diagram shown on Figure 5.2, where transfer function $P_0 = P_0(s)$ is given, the low frequency component of w (below 1 rad/sec) models the reference signal, and the high frequency component of w (above 10 rad/sec) models sensor noise. A standard H-Infinity optimization setup for the task, with two external noise variables w_1, w_2, three cost variables e_1, e_2, e_3, and is shown on Figure 5.3, where r is a small regularization parameter, and W_1, W_2 are stable transfer functions to be chosen appropriately.

MATLAB function `weights1.m`, the essential part of which is shown below, uses both SIMULINK diagrams to design and test a controller.

```
function T=weights1(P0,W1,W2,r)
assignin('base','r',r);                        % export SIMULINK parameters
assignin('base','W2',W2);
assignin('base','W1',W1);
assignin('base','P0',P0);
[a,b,c,d]=linmod('weights1open');              % extract plant model
p=ss(a,b,c,d);
%[k,g]=h2syn(p,1,1);                           % H2 optimization (optional)
[k,g]=hinfsyn(p,1,1);                          % H-infinity optimization
```

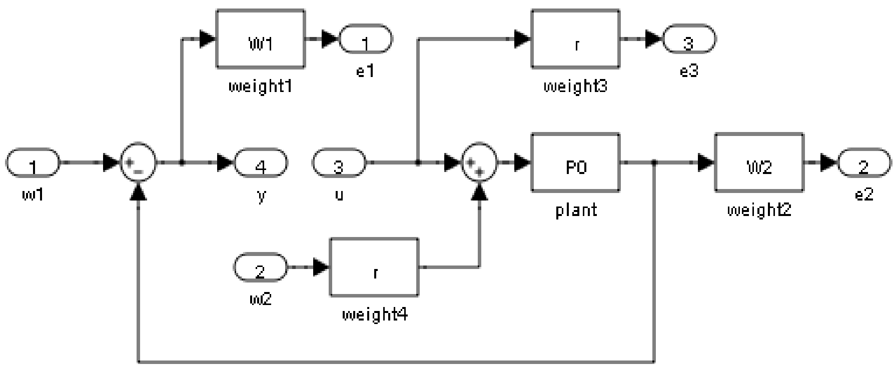

Figure 5.3: Optimization setup with weights

```
assignin('base','K',k);                    % export the controller
[ag,bg,cg,dg]=linmod('weights1closed');    % extract closed loop model
T=ss(ag,bg,cg,dg);
w=logspace(-2,3,500);
Tw=squeeze(freqresp(T,j*w));               % frequency response
close(gcf);semilogx(w,20*log10(abs(1-Tw)));grid
```

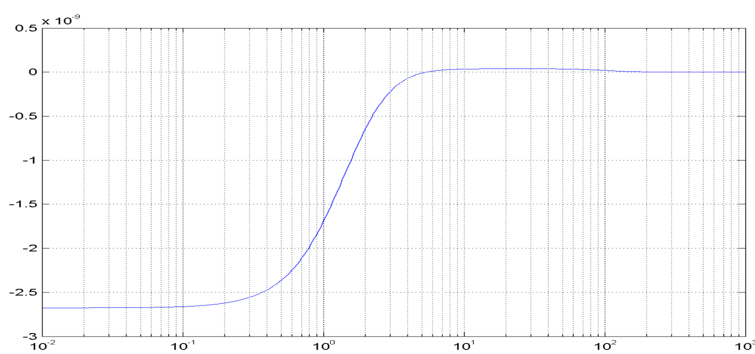

Figure 5.4: Noise rejection for "tough" setup

When used for $P_0(s) = 1/(s+1)^2$ with $r = 0.0001$, $W_2(s) \equiv r$, $W_1(s) \equiv 1$ (i.e. ignoring noise rejection and wishing for good reference tracking across the whole frequency range), H-Infinity optimization produces a puzzling outcome: the closed loop sensitivity transfer function S (from w to $w-e$ on Figure 5.2) is very close to being identically equal to one, which actually means perfect noise rejection and no low frequency reference tracking (see Figure 5.4). The reason for such failure is "toughness" of the setup, a situation when an incorrectly defined cost has a large positive low bound: since P_0 is strictly proper, $S(\infty) = 1$ for every selection of a proper controller K, and hence H-Infinity norm of S can never be made smaller than 1. Since $\|S\|_\infty = 1$ is easy to accomplish (by using $K(s) \equiv 0$), the optimization process "stops there", and does not attempt to make $|S(j\omega)|$ smaller for low frequencies.

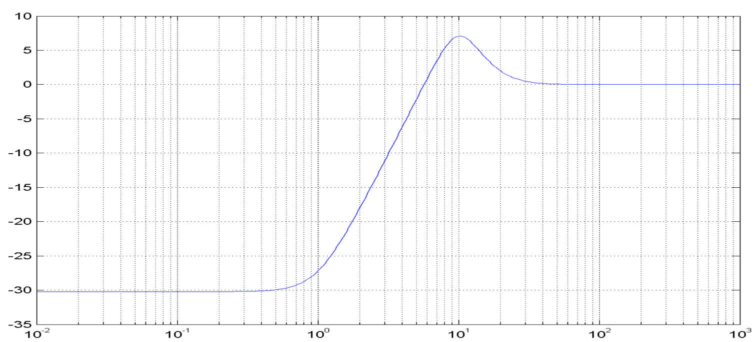

Figure 5.5: Tracking of low frequency signals

To improve low frequency tracking, make the weight W_1 a low-pass transfer function with the cut-off frequency of 1 rad/sec. With $W_1(s) = 1/(s+1)$, optimization yields a third order controller for which reference tracking error drops below -20 dB, while $\|S\|_{\mathcal{L}\infty}$ is well below 1.5. Using a sharper low-pass filter

$$W_1(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

yields a fourth order controller with the Bode plot of S given on Figure 5.5.

To improve high frequency noise rejection, one should use a high-pass filter for $W_2(s)$. However, there is a danger of producing a "tough" setup again. For example, when $W_1(s) = 1/(s+1)$ and $W_2(s) = s/(s+1)$, the low frequency tracking error is about 0.5 (which means a 50 percent error). Indeed, since the transfer matrix from w_1 to $[e_1; e_2]$ equals

$$H(s) = \frac{1}{s+1} \begin{bmatrix} 1 - T(s) \\ sT(s) \end{bmatrix},$$

where $T = 1 - S$ is the complementary sensitivity transfer function, it follows that

$$\sigma_{\max}(H(j)) \geq \frac{(|1 - T(j)|^2 + |T(j)|^2)^{1/2}}{\sqrt{2}} \geq \frac{1}{2}$$

no matter what $T(j)$ is. Once again, the setup turns out to be "tough".

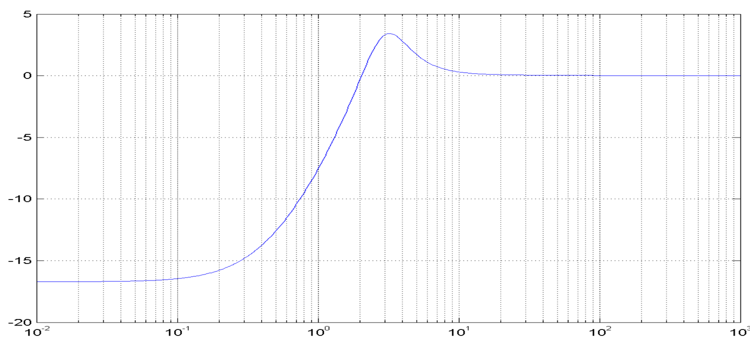

Figure 5.6: Tracking and noise rejection

Feedback optimization with second order Butterworth filters

$$W_1(s) = \frac{1}{s^2 + \sqrt{2}s + 1}, \quad W_2(s) = \frac{s^2}{s^2 + 10\sqrt{2}s + 100},$$

yields a better outcome with both low frequency tracking error and high frequency noise rejection near the -20dB level (see Figure 5.6).

5.3.4 Control Singularity at $\omega = \infty$

Control singularity at $\omega = \infty$ occurs in continuous time feedback optimization setup defined by state space plant equations (5.1)-(5.3) when matrix D_{12} is not left invertible, i.e. when there exists a non-zero vector u_0 such that $D_{12}u_0 = 0$. This case is sometimes referred to as "cheap control", as the control effort along direction u_0 does not enter the cost directly. Accordingly, feedback controller approaching optimality is expected to have a high frequency gain approaching infinity, which means that an optimal proper controller does not exist. To eliminate the singularity, modify the cost variable e by appending a new component, defined as control signal scaled by a small constant.

A very simple example of control singularity at $\omega = \infty$ appears naturally in the problem of stabilizing a first order unstable SISO system when negative unity feedback control is used, and minimization of reference tracking error within the 10 rad/sec bandwidth is desired.

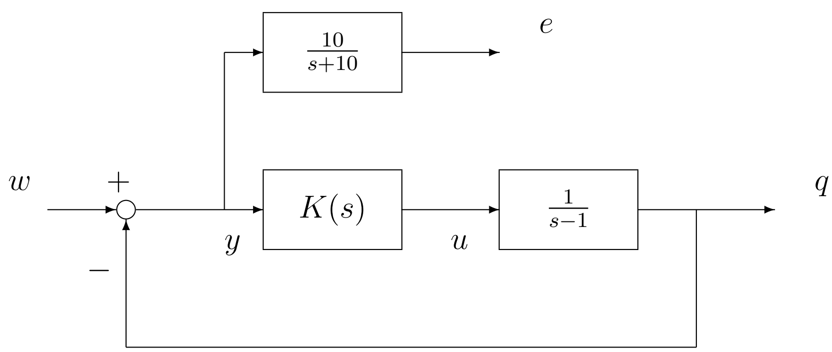

Figure 5.7: Control singularity at $\omega = \infty$

The setup is shown on Figure 5.7. The corresponding plant equations, with the state variable defined as $x = [x_1; x_2]$, where $x_1 = q$ and $x_2 = 0.1e$, are given by

$$\begin{aligned} \dot{x}_1 &= x_1 + u, \\ \dot{x}_2 &= -x_1 - 10x_2 + w, \\ e &= 10x_2, \\ y &= -x_1 + w, \end{aligned}$$

which means that

$$A = \begin{bmatrix} 1 & 0 \\ -1 & -10 \end{bmatrix}, \ B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 0 & 10 \end{bmatrix}, \ C_2 = \begin{bmatrix} -1 & 0 \end{bmatrix}, \ D_{11} = 0, \ D_{12} = 0, \ D_{21} = 1.$$

The control singularity, formally implied by equality $D_{12} = 0$, is also evident in Figure 5.7, as it is clear that the open loop gain "from u to e" is approaching zero at high frequencies.

Using a large positive constant feedback gain $K =$ const yields the closed loop transfer function

$$G(s) = \frac{10}{s+10}\frac{s-1}{s+K-1},$$

(from w to e) which converges to zero in both H2 and H-Infinity norms as $K \to \infty$. It is also clear that no specific controller would produce a closed loop transfer function which is identically zero. Hence an optimal controller does not exist, which is one of the manifestations of singularity.

A script attempting to do H2 optimization exactly as prescribed by Figure 5.7 produces error message "`d12 does not have full column rank`":

```
A=[1 0;-1 -10];B1=[0;1];B2=[1;0];           % define coefficients
C2=[-1 0];D21=1;D22=0;C1=[0 10];D11=0;D12=0;
p=ss(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]);  % plant
nmeas=1;ncon=1;                             % number of sensors/actuators
[k,g]=h2syn(p,nmeas,ncon);                  % h2 optimization
```

In the past, a similar error message would be given by `hinfsyn.m` as well. The current version, though, performs automatically a *regularization* of the setup.

To fix (or *regularize*) the singularity, append ru, where $r \neq 0$ is a constant parameter, to the cost signal as shown on Figure 5.8. It is a good idea to write the modified

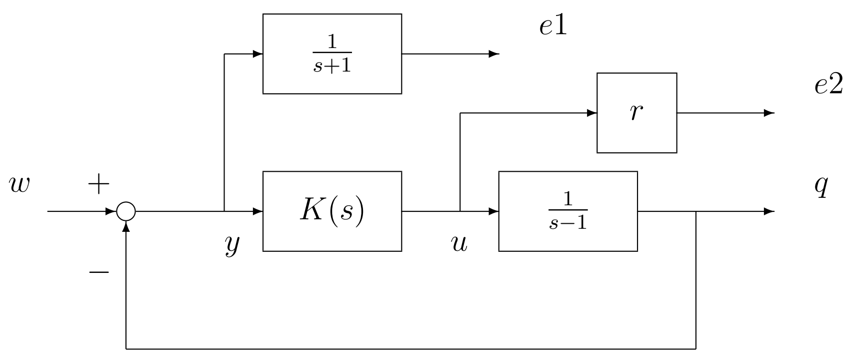

Figure 5.8: Fixing control singularity at $\omega = \infty$

MATLAB code as a function with argument r, to make it easier to observe the effect the regularization parameter has on the closed loop system.

A convenient alternative to defining the coefficient matrices A, B_1, etc. manually is to use SIMULINK diagrams and the linear model extraction function `linmod.m`. For that purpose, build SIMULINK models of open and closed loop system (shown on Figures 5.9 and 5.10). Then MATLAB code

```
s=tf('s');
W=10/(s+10);
P0=1/(s-1);
assignin('base','W',W);                 % assign SIMULINK parameters
```

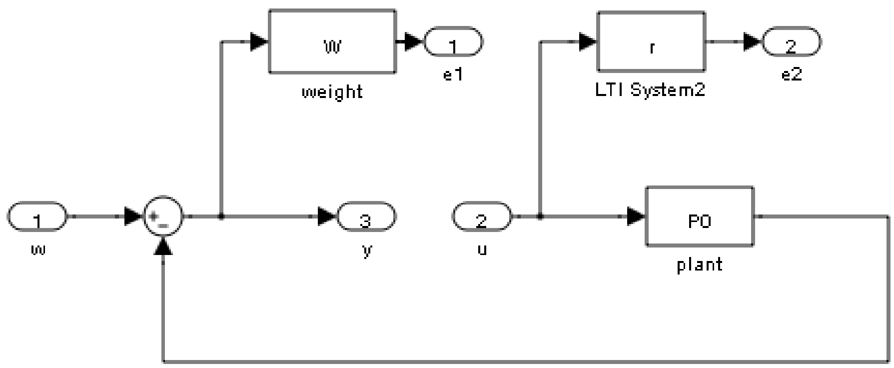

Figure 5.9: SIMULINK model `singular1mod.mdl` of open loop system

```
assignin('base','P0',P0);
assignin('base','r',r);
[ap,bp,cp,dp]=linmod('singular1mod');  % extract plant model
```

extracts the model coefficients from the open loop SIMULINK model, the code

```
p=ss(ap,bp,cp,dp);                     % plant model
nmeas=1;ncon=1;                        % number of sensors/actuators
[k,g]=h2syn(p,nmeas,ncon);             % h2 optimization
```

performs H2 optimization. One can observe from the Bode plot of closed loop sensitivity transfer function S (from w to y, in this case) that, as r converges to zero, the quality of tracking in the 10 rad/sec bandwidth improves, at the cost of increasing the closed loop control gain (w to u).

H-Infinity code for the setup is, essentially, the same.

5.3.5 Sensor Singularity at $\omega = \infty$

Sensor singularity at $\omega = \infty$ occurs in continuous time feedback optimization setup (5.1)-(5.3) when matrix D_{21} is not right invertible, i.e. when there exists a non-zero row vector v_0 such that $v_0 D_{21} = 0$. This means that there is no high frequency noise in the $v_0 y(t)$ component of sensor measurement $y = y(t)$. Accordingly, a feedback controller approaching optimality is expected to use approximate differentiation of $v_0 y(t)$, which means that an optimal proper controller would not exist. To eliminate the singularity, modify the noise variable w by appending a new component w_2, and redefine y as $y + rw_2$, where r a small non-zero constant.

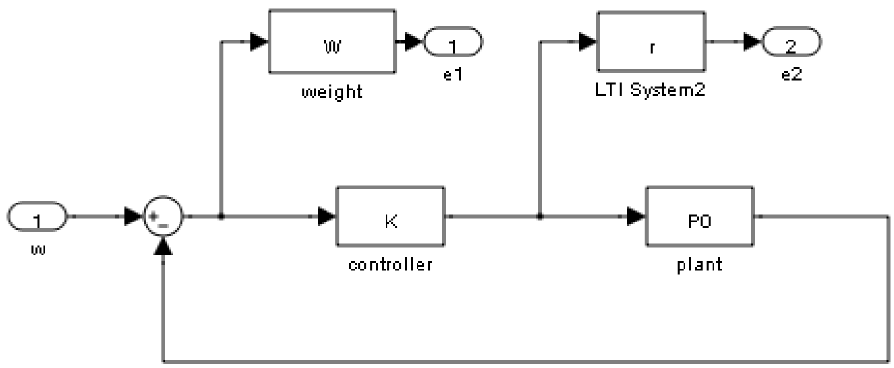

Figure 5.10: SIMULINK model `singular1cmod.mdl` of closed loop system

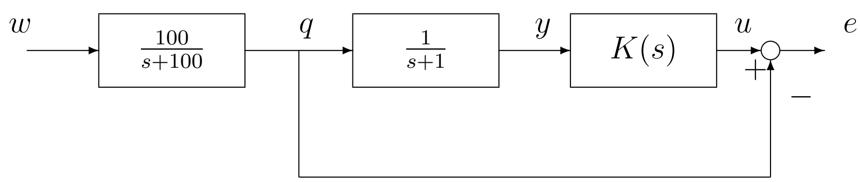

Figure 5.11: Sensor Singularity at $\omega = \infty$

A very simple example of sensor singularity at $\omega = \infty$ appears naturally in the problem of designing a linear estimator for reconstructing a low-pass signal after it has passed through a low-pass filter. The setup is shown on Figure 5.11, where q is the low-pass signal to be reconstructed, K is the estimator to be designed, e is the estimation error, w is assumed to be the "white" noise driving the shaping low-pass filter with cut-off frequency of 100 rad/sec, which models the bandwidth of q. A state space model of the plant, with state vector $x = [x_1; x_2]$, where $x_1 = q$, $x_2 = y$, is given by

$$\begin{aligned} \dot{x}_1 &= -100x_1 + 100w, \\ \dot{x}_2 &= x_1 - x_2, \\ e &= -x_1 + u, \\ y &= x_2, \end{aligned}$$

which yields the coefficient matrices

$$A = \begin{bmatrix} -100 & 0 \\ 1 & -1 \end{bmatrix}, \; B_1 = \begin{bmatrix} 100 \\ 0 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$C_1 = \left[\begin{array}{cc} -1 & 1 \end{array}\right],\ C_2 = \left[\begin{array}{cc} 0 & 1 \end{array}\right],\ D_{11} = D_{22} = D_{21} = 0,\ D_{12} = 1.$$

The sensor singularity at $\omega = \infty$, formally implied by equality $D_{21} = 0$, is also evident in Figure 5.11, as the open loop gain from noise w to sensor y approaches zero at high frequencies.

Using

$$K(s) = \frac{s+1}{\epsilon s + 1},$$

where $\epsilon > 0$ is a pareameter approaching zero, yields closed loop function (from w to e)

$$G(s) = \frac{100}{s+100}\frac{\epsilon s}{s+1}$$

(strictly speaking, there is no real "closed loop" here). Since both H2 and H-Infinity norms of G approach zero as $\epsilon > 0$ approaches zero, an "optimal" $K(s)$ would have to produce $G = 0$. Hence the "optimal" $K(s) = s+1$ is not proper (involves a differentiation operation).

Trying H2 optimization with `h2syn.m` on the setup defined by Figure 5.11 produces an error message: `d21 does not have full row rank`.

```
A=[100,0;1,-1];B1=[100;0];B2=[0;1];         % define coefficients
C1=[-1,0];C2=[0,1];D11=0;D12=1;D21=0;D22=0;
p=ss(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]);  % plant
[k,g]=h2syn(p,1,1,0,100,0.01);              % h2 optimization
```

Latest versions of H-Infinity optimization code regularize the setup automatically, though it is usually a good idea to do your own regularization, to know exactly what is going on. A way to regularize the setup is shown on Figure 5.12, where r should be a small non-zero parameter.

5.3.6 Control Singularity at Finite Frequency

Control singularity at a frequency $\omega = \omega_0 \in \mathbb{R}$ occurs in a feedback optimization setup (5.1)-(5.3) when matrix

$$E_c(s) = \left[\begin{array}{cc} A - sI & B_2 \\ C_1 & D_{12} \end{array}\right]$$

is not left invertible at $s = j\omega_0$ (CT case) or $s = \exp(j\Omega_0)$ (DT case). This typically means that a marginally unstable mode at frequency ω_0 does not show up in the cost of the closed loop system, and thus a controller approaching optimality is expected to

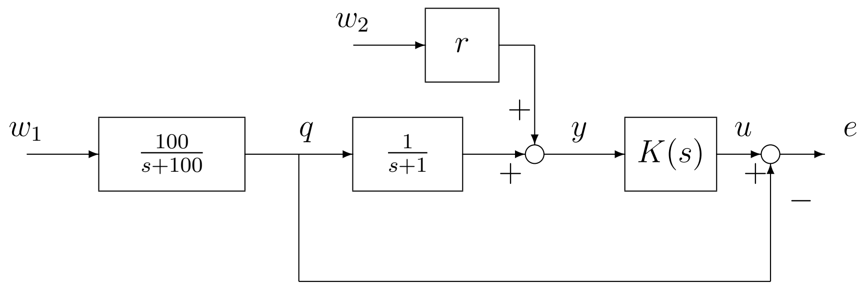

Figure 5.12: Regularized Singularity at $\omega = \infty$

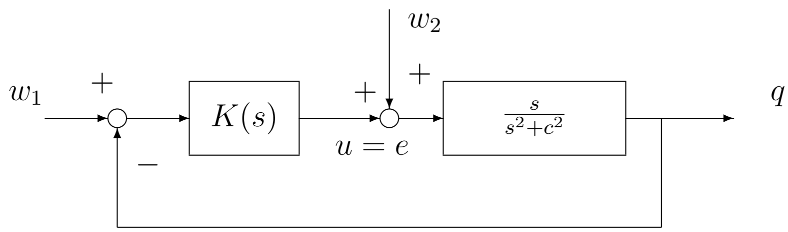

Figure 5.13: CT Control Singularity at $\omega = c$

produce at least one closed loop pole converging to the instability region. To eliminate the singularity, modify the cost variable to include the marginally unstable mode.

A CT setup with control singularity at $\omega = c$ is shown on Figure 5.13, where $c \in \mathbb{R}$ is the resonance frequency of a marginally unstable system disturbed by noise w_2, to be stabilized with minimal control effort by controller K using measurements of q corrupted by noise w_1. A state space model of the plant, with state vector $x = [x_1; x_2]$, where $x_2 = q = \dot{x}_1$ is given by

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= -c^2 x_1 + w_2 + u, \\ e &= u, \\ y &= -x_2 + w_1, \end{aligned}$$

hence

$$A = \begin{bmatrix} 0 & 1 \\ -c^2 & 0 \end{bmatrix}, \; B_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \; B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 0 & 0 \end{bmatrix}, \; C_2 = \begin{bmatrix} 0 & -1 \end{bmatrix}, \; D_{11} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \; D_{12} = 1, \; D_{21} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \; D_{22} = 0.$$

The matrix

$$E_c(s) = \begin{bmatrix} -s & 1 & 0 \\ -c^2 & -s & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

is not invertible for $s = jc$, which formally verifies control singularity of the setup at $\omega = c$. One can also see directly on Figure 5.13 that, though the gain from u to e does not approach zero at frequencies near c, it becomes negligible compared to the gain from u to q.

Functions `h2syn.m` and `hinfsyn.m` are ill-equipped to handle such singularity. H2 optimization in

```
A=[0 1;-1 0];B1=[0,0;0,1];B2=[0;1];          % define coefficients
C1=[0,0];C2=[0,-1];D11=[0,0];D12=1;D21=[1,0];D22=0;
p=ss(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]); % plant
[k,g]=h2syn(p,1,1);                          % optimization
```

is completely screwed up (reports an unintellible `Matrix dimensions must agree`. Replacing the modern formal of `p=ss(...)` by the old Mutools format `p=pck(...)` produces a more informative coded message `Decomposition of X2 failed`, referring to a specific operation in the H2 optimization algorithm. `hinfsyn.m` does (kind of) worse by returning a *non-stabilizing* controller without any warning. Using the old `p=pck(...)` format together with the old style call

```
[k,g]=hinfsyn(p,1,1,0,100,0.01);             % h-infinity optimization
```

returns `Gamma max, 100.0000, is too small !!`, while it can be easily seen that the closed loop H-Infinity norm of G can be made as close to 1 as possible.

To regularize the setup, append rq, where r is a small non-zero constant, to the cost variable.

5.3.7 Sensor Singularity at Finite Frequency

Sensor singularity at a frequency $\omega = \omega_0 \in \mathbb{R}$ occurs in feedback optimization setup (5.1)-(5.3) when matrix

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

is not left invertible at $s = j\omega_0$ (CT case) or $s = \exp(j\Omega_0)$ (DT case). This typically means that a marginally unstable mode at frequency ω_0 is not excited by noise, and thus a controller approaching optimality is expected to use a marginally unstable estimator, producing at least one closed loop pole converging to the instability region. To eliminate the singularity, introduce an extra noise variable exciting the marginally unstable mode.

An example of a CT setup with sensor singularity is shown on Figure 5.14.

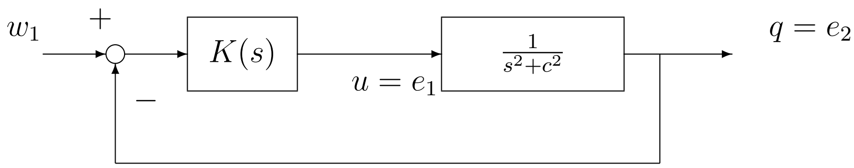

Figure 5.14: Sensor singularity at $\omega = c$