21 Well Posedness of LTI Feedback Design

The standard H2 and H-Infinity optimization tasks are defined by a model of the plant \( P \), a finite order LTI system with input partitioned into noise \( w \) and control \( u \), and output partitioned into cost \( e \) and measurement \( y \):

\[
\dot{x}(t) = Ax(t) + B_1w(t) + B_2u(t), \quad (21.1)
\]
\[
e(t) = C_1x(t) + D_{11}w(t) + D_{12}u(t), \quad (21.2)
\]
\[
y(t) = C_2x(t) + D_{21}w(t). \quad (21.3)
\]

When \( P \) satisfies certain assumptions, relatively simple algorithms are capable of finding a controller \( K \), an finite order LTI system which stabilizes the feedback loop (see Figure 21.1), and either minimizes H2 norm of the closed loop system \( G \) (input \( w \), output \( e \)), or brings H-Infinity norm of \( G \) to within a prescribed level of suboptimality.

![Figure 21.1: Standard feedback design diagram](image)

In applications, some components of the plant model represent physical phenomena describing sensors, actuators, and environment. The corresponding equations are usually obtained through first principles modeling and system identification, not discussed here. In addition, an important part of \( P \) is an artificial construct reflecting the design objectives. While the user of H2 or H-Infinity optimization can do very little to change the true relation between \( u \) and \( y \) (apart from refining the model or re-designing the hardware), it is still possible to generate a state space model in which no stabilizing controller does exist (for example, as a result of careless use of MATLAB transfer matrix to state space transformation routines). In addition, there is a lot of freedom in introducing the cost and noise variables \( e \) and \( w \). When \( w \) and \( e \) are defined incorrectly, one of the two situations typically takes place: either the optimization setup is “unreasonably easy” (singular) due to dangerous omissions and simplifications, or it is “unreasonably hard” (tough) due to
inclusion of unnecessary costs and noises in the formulation. A singular setup happens when one of the matrices $D_{12}$ or

$$W_c(s) = \begin{bmatrix} A - sI & B_2 \\ C_1 & D_{12} \end{bmatrix}, \quad s = j\omega, \ \omega \in \mathbb{R}$$

is not left invertible, or one of the matrices $D_{21}$ or

$$W_m(s) = \begin{bmatrix} A - sI & B_1 \\ C_2 & D_{21} \end{bmatrix}, \quad s = j\omega, \ \omega \in \mathbb{R}$$

is not right invertible, and leads to optimization algorithm failure. A tough setup produces a controller which is optimal with respect to the formal criterion, but does very little to meet the true design objectives.

This section is devoted to studying the causes, consequences, and remedies for non-stabilizability, singularity, and toughness in standard $H_2$ and $H$-Infinity optimization. It also contains several examples of numerical implementation of feedback design in MATLAB.

### 21.1 Lack of Stabilizability

The following statement gives necessary and sufficient conditions of stabilizability of the setup defined by (21.1)-(21.3).

**Theorem 21.1** A controller of the form

$$u(t) = C_f x_f(t) + D_f y(t), \quad \dot{x}_f(t) = A_f x_f(t) + B_f y(t)$$

stabilizing system (21.1)-(21.3) exists if and only if the pair $(A, B_2)$ is stabilizable and the pair $(C_2, A)$ is detectable, i.e. there exist real matrices $F, L$ such that $A + B_2 F$ and $A + L C_2$ are Hurwitz matrices.

**Proof.** If matrices $F, L$ do exist the observer based controller

$$u(t) = F x_f(t), \quad \dot{x}_f(t) = A x_f(t) + B_2 u(t) + L(C_2 x_f(t) - y(t)),$$

in which $x_f(t)$ can be interpreted as an estimate of $x(t)$, stabilizes system (21.1)-(21.3). Indeed, introducing the new state variable $\delta = x - x_f$ allows one to write the closed loop equations in the form

$$\dot{\delta} = (A + L C_2) \delta + (B_1 + L D_{21}) w, \quad \dot{x} = (A + B_2 F) x - B_2 \delta + B_1 w,$$

which is obviously stable.

Assume now that the pair $(A, B_2)$ is not stabilizable. Then the the set $\Omega$ of symmetric $n$-by-$n$ matrices

$$\Omega = \{ Q + AP + PA' + B_2 H + H' B_2' : \ P = P', \ Q = Q', \ H \in \mathbb{R}^{m \times n} \}$$

2
does not contain the zero matrix (otherwise \((A + B_2K)P + P(A + B_2K)' < 0\) for \(K = HP^{-1}\)
which means that \(A + B_2K\) is a Hurwitz matrix). Since \(\Omega\) is a convex set (in the real vector
space \(S_n\) of symmetric \(n\)-by-\(n\) matrices), and \(0 \not\in \Omega\), according to the Hahn-Banach Theorem
there exists a hyperplane such that \(\Omega\) lies on one side of it. Equivalently, there exists a non-zero
symmetric matrix \(Z = Z' \neq 0\) such that
\[
\text{tr } Z(Q + AP + PA' + B_2H + H'B_2') \geq 0 \quad \forall \ Q = Q' > 0, \ P = P' > 0, \ H \in \mathbb{R}^{m \times n},
\]
which implies
\[
Z \geq 0, \quad ZA + A'Z \geq 0, \quad ZB = 0.
\]
Hence
\[
2x'Z(Ax + B_2u) \geq 0 \quad \forall \ x \in \mathbb{R}^n, \ u \in \mathbb{R}^m,
\]
which means that \(V(x(t)) = x(t)'Zx(t)\) is monotonically not decreasing along the solutions of
(21.1), i.e. system (21.1)-(21.3) is not stabilizable.

To show that lack of detectability of the pair \((C_2, A)\) will have a similar effect, note that the
closed loop system is stable if and only if the matrix
\[
A_{cl}[A, B_2, C_2, A_f, B_f, C_f, D_f] \\text{def} = \begin{bmatrix}
A + B_2D_fC_2 & B_2C_f \\
B_fC_2 & A_f
\end{bmatrix}
\]
is a Hurwitz one. Since
\[
A_{cl}[A, B_2, C_2, A_f, B_f, C_f, D_f]' = A_{cl}[A', C_2', B_2', A_f', C_f', B_f', D_f'],
\]
the pair \((A', C_2')\) must be stabilizable. Equivalently, there exists \(F\) such that \(A' + C_2'F\) is a
Hurwitz matrix, and so is \(A + LC_2 = (A' + C_2'F)'\) for \(L = F'\).

At a fundamental level, lack of stabilizability in the pair \((A, B_2)\) should mean that the
actuator hardware is to be reconfigured, either by adding extra devices or by changing
location of the existing ones. Similarly, as a rule, if the pair \((C_2, A)\) is not detectable
then the sensing hardware setup is to be modified. However, there are two important
exceptions to this rule: signal processing tasks for unstable signal models, and silly state
space models resulting from careless use of numerical algorithms.

### 21.1.1 Incorrect State Space Models

Consider the state space feedback stabilization setup given by
\[
\dot{x} = w + u, \quad e = x, \quad y = x.
\]
Here the pair \((A, B_2) = (0, 1)\) is stabilizable, and the pair \((C_2, A) = (1, 0)\) is detectable,
so, as predicted a stabilizing feedback does exist (for example, \(u = -y\)). However, an
attempt to generate a state space model through MATLAB manipulations, as in
\[
s=\text{tf('s'); [A,B,C,D]=ssdata(([1;1]*(1/s)*[1 1]))}
\]
results in
\[ A = D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}. \]

Here the pair \((A, B_2)\) (where \(B_2\) is the second column of \(B\)) is not stabilizable, and the pair \((C_2, A)\) (where \(C_2\) is the second row of \(C\)) is not detectable. Of course, the reason is that, for whatever reason, MATLAB is not capable of producing a correct state space model in this case. Sadly, even
\[
s = \text{tf('s')}; [A, B, C, D] = \text{ssdata}(\text{minreal}([1;1]*(1/s)*[1 1]))
\]
produces the same inadequate output. Only
\[
s = \text{tf('s')}; [A, B, C, D] = \text{ssdata}(\text{minreal}(\text{ss}([1;1]*(1/s)*[1 1])))
\]
fixes the problem.

21.1.2 Estimation of Unstable Dynamics

Consider the setup given by
\[
\dot{x} = w_1, \quad e = x - u, \quad y = x + w_2,
\]
which can be used to design of a dynamical filter which transforms noisy measurements \(y\) of a Brownian motion process \(x\) into its estimate \(u\), trying to minimize the estimation error \(e\). Here the pair \((A, B_2) = (0, 0)\) is not stabilizable, and for a good reason: the plant dynamics is unstable, and no means for feedback are provided. Naturally, an attempt to use \texttt{h2syn.m} to design the optimal filter in this case fails: the code
\[
a = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}; b = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}; c = [1;1]; d = \begin{bmatrix} 0 & 0 & -1; 0 & 1 & 0 \end{bmatrix}; \% \text{state space model}
p = \text{pck}(a, b, c, d); \% \text{convert to old format}
k = \text{h2syn}(p, 1, 1); \% \text{an attempt to optimize}
\]
produces the error message

\texttt{Decomposition of X2 failed}

which is the cryptic way for MATLAB to tell that the pair \((A, B_2)\) was not stabilizable (the same error message may also appear for other reasons). Nevertheless, the task of designing a “controller” system transforming \(y\) into \(u\) to make H2 norm of the transfer function from \(w = [w_1; w_2]\) to \(e\) as small as possible is quite meaningful.

One way to resolve the lack of stabilizability problem is by introducing a non-optimal estimator, such as
\[
\mathcal{K}_o: \quad \dot{q} = -L(q - y),
\]
first (where the \textit{estimator gain} \(L\) must be positive), and then re-formulating the original task as that of designing a correcting filter \(\mathcal{K}_s\) which uses \(y_s = y - q\) as the measurement.
input and produces the output $u_s$, to be added to $q$ to produce the estimate $u = q + u_s$, as shown on Figure 21.2.

The modified setup is given by

$$\dot{x}_s = -Lx_s + w_1 - Lw_2, \quad e_s = x_s - u_s, \quad y_s = x_s + w_2,$$

in which $A = -L$ is a Hurwitz matrix.

MATLAB’s `h2syn.m` can be used to design the optimal filter using the following code:

```matlab
L=3; % rough estimator gain
a=-L; b=[1 -L 0]; c=[1;1]; d=[0 0 -1;0 1 0]; % state space model
p=pck(a,b,c,d); % convert to old format
k=h2syn(p,1,1); % an attempt to optimize
[A,B,C,D]=unpck(k);Ks=ss(A,B,C,D); % convert to state space
tf(Ks) % correcting filter
Ko=ss(-L,L,1,0); % original filter
K=minreal(Ko+Ks*(1-Ko)); tf(K) % optimal filter
```

which produces transfer

$$K_o(s) = \frac{3}{s+3}, \quad K_s(s) = -\frac{2}{s+1}, \quad K(s) = \frac{1}{s+1}.$$  

Note that, as expected, the total optimal filter has order 1, and does not depend on $L$.

### 21.2 Control Singularity at $\omega = \infty$

Control singularity at $\omega = \infty$ occurs in continuous time feedback optimization setup defined by state space plant equations (21.1)-(21.3) 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.

The setup is shown on Figure 21.3. 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

\[
\begin{align*}
\dot{x}_1 &= x_1 + u, \\
\dot{x}_2 &= -x_1 - 10x_2 + w, \\
e &= 10x_2, \\
y &= -x_1 + w,
\end{align*}
\]

which means that

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

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

The control singularity, formally implied by equality $D_{12} = 0$, is also evident in Figure 21.3, 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 $K = \text{const}$ yields the closed loop transfer function

\[
G(s) = \frac{10}{s + 10} \cdot \frac{s - 1}{s + K - 1},
\]

(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 21.3 produces error message “d12 does not have full column rank”:

\[
A = \begin{bmatrix} 1 & 0; -1 & -10 \end{bmatrix}; B1 = [0; 1]; B2 = [1; 0]; \]
\[
C2 = [-1 & 0]; D21 = 1; D22 = 0; C1 = [0 & 10]; D11 = 0; D12 = 0; \]
\[
p = \text{pck}(A, [B1 B2], [C1; C2], [D11 D12; D21 D22]); \]
\[
nmeas = 1; ncon = 1; \]
\[
[k, g] = \text{h2syn}(p, nmeas, ncon); \]

A similar error message is given by hinfsyn.m.

To fix (or regularize) the singularity, append \( ru \), where \( r \neq 0 \) is a constant parameter, to the cost signal as shown on Figure 21.4. It is a good idea to write the modified 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.

\[
\begin{align*}
&\frac{1}{s+1} & e1 \\
\end{align*}
\]

\[
\begin{align*}
&K(s) \quad \frac{1}{s-1} \\
\end{align*}
\]

Figure 21.4: 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.

\[
\begin{align*}
&w & e1 \\
\end{align*}
\]

\[
\begin{align*}
&y & u \\
\end{align*}
\]

\[
\begin{align*}
&q & e2 \\
\end{align*}
\]

Figure 21.5: SIMULINK model singular1mod.mdl of open 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 \texttt{linmod.m}. For that purpose, build SIMULINK models of open and closed loop system (shown on Figures 21.5 and 21.6). Then MATLAB code

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

```matlab
p=pck(ap,bp,cp,dp); % plant model in mutools format
mmeas=1;ncon=1; % number of sensors/actuators
[k,g]=h2syn(p,mmeas,ncon); % h2 optimization
[ak,bk,ck,dk]=unpck(k); % unpack controller coef
```

performs H2 optimization. A complete H2 optimization code for this setup is available in \texttt{singular1.m}. 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/see bandwidth improves, at the cost of increasing the closed loop control gain (\( w \) to \( u \)).

![Diagram](image)

Figure 21.6: SIMULINK model \texttt{singular1cmod.mdl} of closed loop system

H-Infinity code for the setup is, essentially, the same, except that \texttt{hinfsyn.m} requires additional parameters, lower and upper bounds \( \text{gmin}, \text{gmax} \), and the suboptimality tolerance \( \text{tol} \) for the binary search:

```matlab
mmeas=1;ncon=1;gmin=0;gmax=100;tol=0.01; % parameters of hinfsyn
[k,g]=hinfsyn(p,mmeas,ncon,gmin,gmax,tol); % h-infinity optimization
```
21.3 Sensor Singularity at \( \omega = \infty \)

Sensor singularity at \( \omega = \infty \) occurs in continuous time feedback optimization setup (21.1)-(21.3) when matrix \( D_{21} \) is not right invertible, i.e. when there exists a non-zero row vector \( v_0 \) such that \( v_0D_{21} = 0 \). This means that there is no high frequency noise in the \( v_0y(t) \) component of sensor measurement \( y = y(t) \). Accordingly, a feedback controller approaching optimality is expected to use approximate differentiation of \( v_0y(t) \), 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 21.7: Sensor Singularity at \( \omega = \infty \)](image)

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 21.7, 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

\[
\begin{align*}
\dot{x}_1 &= -100x_1 + 100w, \\
\dot{x}_2 &= x_1 - x_2, \\
e &= -x_1 + u, \\
y &= x_2,
\end{align*}
\]

which yields the coefficient matrices

\[
A = \begin{bmatrix}
-100 & 0 \\
1 & -1
\end{bmatrix}, \quad B_1 = \begin{bmatrix}
100 \\
0
\end{bmatrix}, \quad B_2 = \begin{bmatrix}
0 \\
1
\end{bmatrix},
\]

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

The sensor singularity at \( \omega = \infty \), formally implied by equality \( D_{21} = 0 \), is also evident in Figure 21.7, as the open loop gain from noise \( w \) to sensor \( y \) approaches zero at high frequencies.
Using
\[ K(s) = \frac{s + 1}{\varepsilon s + 1}, \]
where \( \varepsilon > 0 \) is a parameter approaching zero, yields closed loop function (from \( w \) to \( e \))
\[ G(s) = \frac{100 \varepsilon s}{s + 100 s + 1} \]
(strictly speaking, there is no real “closed loop” here). Since both H2 and H-Infinity norms of \( G \) approach zero as \( \varepsilon > 0 \) 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 H-Infinity optimization with \texttt{hinfsyn.m} on the setup defined by Figure 21.7 produces an error message which containing \( d21 \) does not have full row rank preceded by some garbage.

\begin{verbatim}
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=pck(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]);  % plant in mutools format
[k,g]=hinfsyn(p,1,1,0,100,0.01);             % h-infinity optimization
\end{verbatim}

H2 optimization with \texttt{h2syn.m} yields the same info minus the dump. A way to regularize the setup is shown on Figure 21.8, where \( r \) should be a small non-zero parameter.

![Figure 21.8: Regularized Singularity at \( \omega = \infty \)](image)

### 21.4 Control Singularity at Finite Frequency

Control singularity at a frequency \( \omega = \omega_0 \in \mathbb{R} \) occurs in continuous time feedback optimization setup (21.1)-(21.3) when matrix
\[ E_c(s) = \begin{bmatrix} A - sI & B_2 \\ C_1 & D_{12} \end{bmatrix} \]
is not left invertible at \( s = j\omega_0 \). Control singularity at a frequency \( \omega = \omega_0 \in [0, \pi] \) occurs in discrete time feedback optimization setup

\[
x(t + 1) = Ax(t) + B_1 w(t) + B_2 u(t), \tag{21.4}
\]

\[
e(t) = C_1 x(t) + D_{11} w(t) + D_{12} u(t), \tag{21.5}
\]

\[
y(t) = C_2 x(t) + D_{21} w(t), \tag{21.6}
\]

when \( E_c(z) \) is not left invertible at \( z = e^{j\omega_0} \). This typically means that a marginally unstable mode at frequency \( \omega_0 \) does not show up in the cost of the closed loop system, and thus a controller approaching optimality is expected to 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.

![CT Control Singularity at \( \omega = \omega_0 \)](image)

**Figure 21.9: CT Control Singularity at \( \omega = \omega_0 \)**

A CT setup with control singularity at \( \omega = \omega_0 \) is shown on Figure 21.9, 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

\[
\dot{x}_1 = x_2,
\]

\[
\dot{x}_2 = -c^2 x_1 + w_2 + u,
\]

\[
e = u,
\]

\[
y = -x_2 + w_1,
\]

hence

\[
A = \begin{bmatrix} 0 & 1 \\ -c^2 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \end{bmatrix},
\]

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

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 21.9 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

$$A=[0 \ 1; -1 \ 0]; B1=[0,0;0,1]; B2=[0;1];$$
$$C1=[0,0]; C2=[0,-1]; D11=[0,0]; D12=1; D21=[1,0]; D22=0;$$
$$p=pck(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]);$$
$$[k,g]=h2syn(p,1,1);$$

produces coded message Decomposition of X2 failed, while hinfsyn.m goes further by declaring in

$$[k,g]=hinfsyn(p,1,1,0,100,0.01);$$

that 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.

### 21.5 Sensor Singularity at Finite Frequency

Sensor singularity at a frequency $\omega = \omega_0 \in \mathbb{R}$ occurs in continuous time feedback optimization setup (21.1)-(21.3) when matrix

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

is not left invertible at $s = j\omega_0$. Sensor singularity at a frequency $\omega = \omega_0 \in [0, \pi]$ occurs in discrete time feedback optimization setup (21.4)-(21.6) when $E_m(z)$ is not left invertible at $z = e^{j\omega_0}$. This typically means that a marginally unstable mode at frequency $\omega_0$ 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 21.10.
Figure 21.10: Sensor singularity at $\omega = c$