

Massachusetts Institute of Technology

Department of Electrical Engineering and Computer Science
6.245: MULTIVARIABLE CONTROL SYSTEMS

by A. Megretski *

Standard LTI Feedback Optimization Setup¹

The two most frequently used techniques of optimization-based feedback design are H₂ and H-∞ optimization. Both are normally applied to what can be referred to as the *standard LTI feedback optimization setup*, to be the main subject of discussion in this lecture.

1.1 Basic Concepts

There are three basic concepts behind the standard feedback optimization setup: the notion of a multi-input, multi-output (MIMO) linear time-invariant (LTI) finite order system, the notion of an *internally stable feedback interconnection* of two MIMO LTI systems, and the notion of a *system norm*. In addition, there is a concept of *well-posedness* of the resulting optimization problem, which has to be satisfied (otherwise the available optimization algorithms are guaranteed to break down).

1.1.1 Components of a standard setup

In order to define a standard LTI feedback optimization problem, sketched on Figure fig:PKstandard, one has to specify the *plant* P , and a performance measure.

*©A. Megretski, 2004

¹Version of February 4, 2004

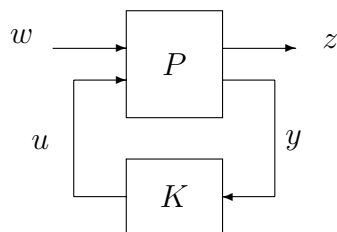


Figure 1.1: Standard LTI feedback optimization setup

The *plant* is a finite order linear time-invariant system with two inputs, w and u , and two outputs, z and y . Inputs and outputs can be multi-variable. The first input w is called *disturbance*, and is typically used to describe external noises and internal perturbations caused by nonlinearity and uncertainty. The second input u is called *actuator*, and describes the output of the feedback controller K (to be designed via optimization). The first output z is called *cost*, and represents signals which the designer wants to be small. Finally, the second output y , usually referred to as *sensor*, or *measurement*, represents input of the controller K (to be designed).

The *performance measure* specifies a particular quantitative measure of “smallness” for the cost output z . The optimization process will aim at finding an LTI feedback system K which makes the feedback system on Figure fig:PKstandard *stable*, and “minimizes” the closed loop system from w to z . Two measures (also called *norms*) of how “large” a stable LTI system is are most popular in feedback optimization: *H2 norm* and *H-Infinity norm*. The H2 norm measures the size of an LTI system as an integral of square of amplitude of its frequency response, while the H-Infinity norm uses the *maximal* (over all frequencies) amplitude.

There are plants P for which the standard H2 and H-Infinity optimization algorithms are guaranteed to fail. Certain *well-posedness* conditions (to be discussed later in this lecture) have to be satisfied to avoid such failures.

1.1.2 State-space format for continuous time LTI systems

While H2 and H-Infinity optimization can be applied to discrete-time (DT) and continuous-time (CT) systems, represented in many of possible formats (state-space, transfer matrix, zero-pole, etc.), the basic algorithms are traditionally developed and explained for CT systems defined by state-space equations. A state-space model for a finite order CT LTI

system with input $f = f(t)$, output $g = g(t)$, and state $x = x(t)$ has the form

$$\dot{x}(t) = Ax(t) + Bf(t), \quad (1.1)$$

$$g(t) = Cx(t) + Df(t), \quad (1.2)$$

where A, B, C, D are constant matrices with real entries. In the modern control literature, it is common to use the notation

$$\mathcal{S} = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right)$$

for this system. For example, the SISO CT LTI system with transfer function $G(s) = 1/s^2$ can be represented by a state space model with

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

When a CT LTI system describes the plant in a standard feedback optimization setup, its input is partitioned into the disturbance and actuator components. Similarly, the output is partitioned into the cost and measurement components. Consequently, it is natural to decompose the corresponding B, C, D matrices:

$$\dot{x} = Ax + B_1w + B_2u, \quad (1.3)$$

$$z = C_1x + D_{11}w + D_{12}u, \quad (1.4)$$

$$y = C_2x + D_{21}w + D_{22}u, \quad (1.5)$$

or

$$P = \left(\begin{array}{c|c} A & [B_1 \quad B_2] \\ \hline [C_1] & [D_{11} \quad D_{12}] \\ [C_2] & [D_{21} \quad D_{22}] \end{array} \right).$$

A state-space model of controller K (which is an LTI system transforming y into u) can be given in the form

$$\dot{x}_f = A_fx_f + B_fy \quad (1.6)$$

$$u = C_fx_c + D_fy \quad (1.7)$$

where A_f, B_f, C_f, D_f are matrices with real coefficients, to be found by an optimization algorithm.

1.1.3 Well-posedness

There are three sets of constraints which have to be imposed on a standard LTI feedback optimization setup for it to be well-posed and numerically well conditioned.

The first set of constraints guarantees *existence* of a stabilizing feedback (so that the set of feasible decision parameters is not empty). This requires the pair (A, B_2) to be stabilizable, and the pair (C_2, A) to be detectable. If this is not the case, the *physical* feedback control setup should be modified by adding extra actuators (to make the pair (A, B_2) stabilizable) and/or sensors (to make the pair (C_2, A) detectable).

The second set of constraints guarantees *existence* of an optimal controller (and is also related to numerical well-posedness of the optimization problem). Informally speaking, it requires the setup to be “hard enough”, in the sense that every component of the measurement output of the plant should be dependent on the disturbance input *at every frequency*, and every component of the actuator input should affect the cost output *at every frequency*. The formal constraints are as follows: matrix

$$M_u(s) = \begin{bmatrix} A - sI & B_2 \\ C_1 & D_{12} \end{bmatrix} \quad (1.8)$$

must be left invertible (i.e. its columns must be linearly independent) for all s on the imaginary axis, including the case $s = \infty$, when the condition is that matrix D_{12} must be left invertible; similarly, matrix

$$M_y(s) = \begin{bmatrix} A - sI & B_1 \\ C_2 & D_{21} \end{bmatrix} \quad (1.9)$$

must be right invertible (i.e. its rows must be linearly independent) for all s on the imaginary axis, including the case $s = \infty$, when the condition is that matrix D_{21} must be right invertible. When A has no eigenvalues on the imaginary axis, the condition can be re-written as left invertibility of transfer matrix

$$P_{12}(s) = P_{u \rightarrow z}(s) = C_1(sI - A)^{-1}B_2 + D_{12},$$

and right invertibility of

$$P_{21}(s) = P_{w \rightarrow y}(s) = C_2(sI - A)^{-1}B_1 + D_{21}.$$

The case when $M_u(s)$ is not left invertible at some $s = j\omega$ will be referred to as *control singularity at frequency ω* . Similarly, the case when $M_y(s)$ is not right invertible at some $s = j\omega$ will be referred to as *sensor singularity at frequency ω* .

Finally, the third set of constraints, applicable to H2 optimization only, requires $D_{11} = 0$ and $D_{22} = 0$. Every setup with $D_{22} \neq 0$ can be reduced easily to the case with $D_{22} = 0$ (any direct presence of control signal in the sensor signal can be compensated, since u is “known” anyway. An H2 optimization setup with $D_{11} \neq 0$ is not necessarily meaningless, it is just not supported by the standard H2 optimization software. After learning about the inner workings of H2 optimization, you will be able to upgrade the existing algorithms to the case $D_{11} \neq 0$.

1.1.4 MATLAB format for H2 and H-Infinity optimization

This subsection describes the use of Mu-Analysis and Synthesis Toolbox, the recommended set of routines for H2 and H-Infinity optimization.

To define a plant state space model P in this toolbox, use

```
P=pck(A,B,C,D);
```

where A, B, C, D are the matrices defining

$$P = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right).$$

To call an H2 optimization algorithm with a minimal set of input and output arguments, use

```
K=h2syn(P,nmeas,ncon);
```

where P is the plant model in the packed format, **nmeas** is the number of sensors (i.e. the dimension of vector y), **ncon** is the number of actuators (the dimension of u), K is the optimal controller model in the packed format. To get the coefficient matrices of the controller, unpack K with

```
[Af,Bf,Cf,Df]=unpck(K);
```

To call an H2 optimization algorithm with a minimal set of input and output arguments, use

```
K=hinfsyn(P,nmeas,ncon,gmin,gmax,tol);
```

Here the output argument and the first three input arguments have same meaning as in H2 optimization. The presence of the last three input arguments is caused by the fact that function `hinfsyn.m` is not capable of finding the *optimal* H-Infinity controller.

Instead it searches for a controller which yields closed loop H-Infinity norm γ , such that $(\gamma - \gamma_{\min})/\gamma$ is not larger than the relative tolerance parameter `tol`. Here γ_{\min} is the minimal achievable H-Infinity norm, `gmin` is a known lower bound for γ_{\min} (one can safely use `gmin=0`), and `gmax` is a known upper bound for γ_{\min} . While it is not always easy to find an upper bound, `hinfsyn.m` will tell you if your current guess is too low.

1.1.5 A simple design example

Consider a simple feedback design task shown on Figure fig:PFsimple, where $G = G(s) = 1/s^2$ is a given open loop plant model, and $F = F(s)$ is the feedback controller to be designed to provide a desired closed loop response $T = T(s)$ from reference input r to controlled output q . Assume that the “ideal” desired closed loop response is $T(s) =$

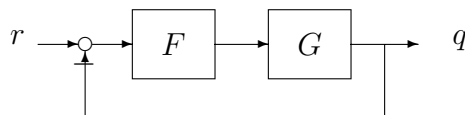


Figure 1.2: A simple feedback design example

$T_0(s) = 1/(s + 1)$. This response cannot be achieved by using a proper controller transfer function $F(s)$. However, one can try to approximate the ideal response $T_0(s)$ by choosing an appropriate stabilizing proper controller F , while checking the trade-off between the quality of approximation and the power utilized by the controller. Both H2 and H-Infinity optimization frameworks are easy to use for this purpose.

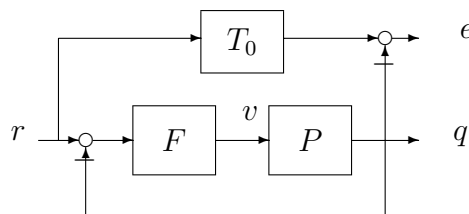


Figure 1.3: Reduction to standard optimization setup

To re-write the design specifications as a standard feedback optimization setup, introduce the ideal response transfer function to the block diagram, and define e as the difference between the actual and the desired response. Our objective is to make the closed loop transfer function from r to e “small” by selecting a controller with input $r - q$

and output v . In terms of the standard setup, this calls for selecting r as the disturbance input w , v as the actuator input u , $r - q$ as the sensor output y , and e as the cost output z . The corresponding plant transfer matrix P is given by

$$P(s) = \begin{bmatrix} 1/(s+1) & -1/s^2 \\ 1 & -1/s^2 \end{bmatrix},$$

which corresponds to a minimal state space model with

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

$$C_1 = [1 \ 0 \ -1], \quad C_2 = [0 \ 0 \ -1], \quad D_{11} = D_{22} = 0, \quad D_{12} = 0, \quad D_{21} = 1.$$

An attempt to use `h2syn.m` or `hinfsv.m` on this setup will produce an error message, because the setup is not well-posed. One obvious reason for this is absence of a control penalty (causing a control singularity at $\omega = \infty$). This can be fixed by adding $\epsilon_u v$ as an extra component of the cost z , where ϵ_u will become a tuning parameter for the designer (the larger ϵ_u is, the less power the optimal controller will use, at the expense of providing a poorer approximation of the desired closed loop response).

A less obvious problem with the setup is a sensor singularity at $\omega = 0$, which is not as easy to spot since the open loop plant has a pole at $s = 0$. Actually, this is a *double* sensor singularity at $\omega = 0$, since the determinant of $M_y(s)$ has a double root at $s = 0$. This singularity can be fixed by having an extra disturbance signal $f = \epsilon_y w_2$ added to the input of the double integrator. Here parameter ϵ_y will quantify sensitivity of the closed loop system with respect to the plant disturbances (the smaller ϵ_y , the larger the sensitivity).

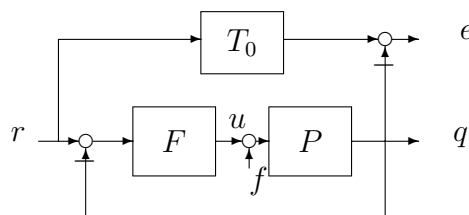


Figure 1.4:

The resulting standard setup will have two-component w , two-component z , and two

tuning parameters ϵ_u and ϵ_y . The plant transfer matrix will have the form

$$P(s) = \begin{bmatrix} 1/(s+1) & -\epsilon_y/s^2 & -1/s^2 \\ 0 & 0 & \epsilon_u \\ 1 & -\epsilon_y/s^2 & -1/s^2 \end{bmatrix},$$

and a minimal state space model given by

$$\dot{x}_1(t) = -x_1(t) + w_1(t), \quad (1.10)$$

$$\dot{x}_2(t) = u(t) + \epsilon_y w_2(t), \quad (1.11)$$

$$\dot{x}_3(t) = x_2(t), \quad (1.12)$$

$$z_1(t) = x_1(t) - x_3(t), \quad (1.13)$$

$$z_2(t) = \epsilon_u u(t), \quad (1.14)$$

$$y(t) = -x_3(t) + w_1(t). \quad (1.15)$$

A sample MATLAB code for H2 optimization is given by

```
eu=0.01;
ey=0.01;
A=[-1 0 0;0 0 0;0 1 0];
B1=[1 0;0 ey;0 0];
B2=[0;1;0];
C1=[1 0 -1;0 0 0];
C2=[0 0 -1];
D11=zeros(2);
D12=[0;eu];
D21=[1 0];
D22=0;
P=pck(A,[B1 B2],[C1;C2],[D11 D12;D21 D22]);
K=h2syn(P,1,1);
[Af,Bf,Cf,Df]=unpck(K);
```

To modify this for H-Infinity optimization, simply replace the `h2syn` line with

```
K=hinfsyn(P,1,1,0,1,0.01);
```