Undergraduate Lecture Notes on

LQG/LQR controller design

João P. Hespanha

April 1, 2007

Disclaimer: This is a draft and probably contains several typos.

Comments and information about typos are welcome. Please contact the author at (`hespanha@ece.ucsb.edu`).

Contents

Attention! When a margin sidebar finishes with "...", more information about the topic can be found at the end of the lecture in the "To probe further" section.

Introduction to LQG/LQR controller design

In *optimal control* one attempts to find a controller that provides the best possible performance with respect to some given *measure of performance.* E.g., the controller that uses the least amount of control-signal energy to take the output to zero. In this case the measure of performance (also called the *optimality criterion*) would be the control-signal energy.

In general, optimality with respect to some criterion is not the only desirable property for a controller. One would also like stability of the closed-loop system, good gain and phase margins, robustness with respect to unmodeled dynamics, etc.

In this section we study controllers that are optimal with respect to energy-like criteria. These are particularly interesting because the minimization procedure *automatically produces controllers that are stable and somewhat robust.* In fact, the controllers obtained through this procedure are generally so good that we often use them *even when we do not necessarily care about optimizing for energy.* Moreover, this procedure is applicable to *multiple-input/multiple-output* processes for which classical designs are difficult to apply.

Pre-requisites

1. Basic knowledge of state-space models (briefly reviewed here)
2. Familiarity with basic vector and matrix operations.
3. Knowledge of MATLAB/Simulink.

Lecture #1

Review of State-space models

Contents

1.1 State-space models

Consider the system in Figure 1.1 with m inputs and k outputs. A *state-space* model for

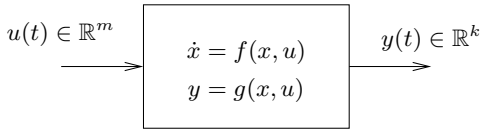


Figure 1.1. System with m inputs and k outputs

this system relates the input and output of a system using the following first-order vector ordinary differential equation

$$\dot{x} = f(x,u), \qquad y = g(x,u). \tag{1.1}$$

where $x \in \mathbb{R}^n$ is called the state of system. In this Chapter we restrict our attention to *linear time-invariant (LTI)* systems for which the functions $f(\cdot,\cdot)$ and $g(\cdot,\cdot)$ are linear. In this case, (1.1) has the special form

$$\dot{x} = Ax + Bu, \qquad y = Cx + Du, \tag{1.2}$$

where A is a $n \times n$ matrix, B a $n \times m$ matrix, C a $k \times n$ matrix, and D a $k \times m$ matrix.

Matlab hint 1. `ss(A,B,C,D)` creates a LTI state-space model with realization (1.2)...

Example 1 (Aircraft roll-dynamics). Figure 1.2 shows the roll-angle dynamics of an aircraft [2, p. 381]. Defining

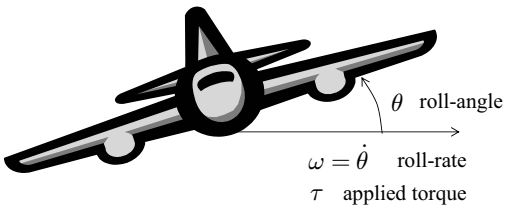


Figure 1.2. Aircraft roll-angle dynamics

$$x := \begin{bmatrix} \theta & \omega & \tau \end{bmatrix}'$$

we conclude that

$$\dot{x} = Ax + Bu$$

with

$$A := \begin{bmatrix} 0 & 1 & 0 \\ 0 & -.875 & -20 \\ 0 & 0 & -50 \end{bmatrix}, \qquad B := \begin{bmatrix} 0 \\ 0 \\ 50 \end{bmatrix}.$$

If we have both θ and ω available for control, we can define

$$y := \begin{bmatrix} \theta & \omega \end{bmatrix}' = Cx + Du$$

with

$$C := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \qquad D := \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \qquad \square$$

1.2 Input-output relations

Sidebar 1. See [1, Appendix A] for a review of Laplace transforms.

The *transfer-function* of this system can be found by taking Laplace transforms of (1.2):

$$\begin{cases} \dot{x} &= Ax + Bu, \\ y &= Cx + Du, \end{cases} \quad \xrightarrow{\mathcal{L}} \quad \begin{cases} sX(s) &= AX(s) + BU(s), \\ Y(s) &= CX(s) + DU(s), \end{cases}$$

where $X(s)$, $U(s)$, and $Y(s)$ denote the Laplace transforms of $x(t)$, $u(t)$, and $y(t)$. Solving for $X(s)$, we get

$$(sI - A)X(s) = BU(s) \quad \Leftrightarrow \quad X(s) = (sI - A)^{-1}BU(s)$$

and therefore

$$Y(s) = C(sI - A)^{-1}BU(s) + DU(s) = \big(C(sI - A)^{-1}B + D\big)U(s).$$

Defining

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

we conclude that

$$Y(s) = T(s)\,U(s). \tag{1.3}$$

To emphasize the fact that $T(s)$ is a $k \times m$ matrix, we call it the *transfer-matrix* of the system (1.2).

The relation (1.3) between the Laplace transforms of the input and the output of the system is only valid for zero initial conditions, i.e., when $x(0) = 0$. The general solution to the system (1.2) in the *time domain* is given by

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-s)}Bu(s)ds, \tag{1.4}$$

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-s)}Bu(s)ds + Du(t), \qquad \forall t \geq 0. \tag{1.5}$$

Equation (1.4) is called the *variation of constants formula*.

Example 2 (Aircraft roll-dynamics). The transfer-function for the state-space model in Example 1 is given by:

$$T(s) = \begin{bmatrix} \frac{-1000}{s(s+.875)(s+50)} \\ \frac{-1000}{(s+.875)(s+50)} \end{bmatrix}$$

□

1.3 Realizations

Consider a transfer-matrix

$$T(s) = \begin{bmatrix} T_{11}(s) & T_{12}(s) & \cdots & T_{1m}(s) \\ T_{21}(s) & T_{22}(s) & \cdots & T_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ T_{k1}(s) & T_{k2}(s) & \cdots & T_{km}(s) \end{bmatrix},$$

where all the $T_{ij}(s)$ are given by a ratio of polynomials with the degree of the numerator smaller than or equal to the degree of the denominator. It is always possible to find matrices A, B, C, D such that

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

This means that it is always possible to find a state-space model like (1.2) whose transfer-matrix is precisely $T(s)$. The model (1.2) is called a *realization* of $T(s)$.

Matlab hint 2. `tf(num,den)` creates a transfer-function with numerator and denominator specified by `num`, `den`...

Matlab hint 3. `zpk(z,p,k)` creates a transfer-function with zeros, poles, and gain specified by `z`, `p`, `k`...

Matlab hint 4. `tf(sys_ss)` and `zpk(sys_ss)` compute the transfer-function of the state-space model `sys_ss`...

Matlab hint 5. `expm` computes the exponential of a matrix...

Matlab hint 6. `ss(sys_tf)` computes a realization of the transfer-function `sys_tf`...

Attention! Realizations are not unique, i.e., several state-space models may have the same transfer function.

1.4 Controllability and observability

The system (1.2) is said to be *controllable* when given any initial state $x_i \in \mathbb{R}^n$, any final state $x_f \in \mathbb{R}^n$, and any finite time T, one can find an input signal $u(t)$ that takes the state of (1.2) from x_i to x_f in the interval of time $0 \leq t \leq T$, i.e., when there exists an input $u(t)$ such that

$$x_f = e^{AT}x_i + \int_0^T e^{A(T-s)}Bu(s)ds.$$

Matlab hint 7. `ctrb(sys)` computes the controllability matrix of the state-space system `sys`. Alternatively, one can use directly `ctrb(A,B)`...

To determine if a system is controllable, one can compute the *controllability matrix*, which is defined by

$$\mathcal{C} := \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}.$$

Matlab hint 8. `rank(M)` computes the rank of a matrix `M`.

The system is controllable if and only if this matrix has rank equal to the size n of the state vector.

The system (1.2) is said to be *observable* when one can determine the initial condition $x(0)$ by simply looking at the input and output signals $u(t)$ and $y(t)$ on a certain interval of time $0 \leq t \leq T$, i.e., one can solve

$$y(t) = Ce^{At}x(0) + \int_0^t Ce^{A(t-s)}Bu(s)ds + Du(t), \qquad \forall\, 0 \leq t \leq T,$$

Matlab hint 9. `obsv(sys)` computes the controllability matrix of the state-space system `sys`. Alternatively, one can use directly `obsv(A,C)`...

uniquely for the unknown $x(0)$. To determine if a system is observable, one can compute the *observability matrix*, which is defined by

$$\mathcal{O} := \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

The system is observable if and only if this matrix has rank equal to the size n of the state vector.

Example 3 (Aircraft roll-dynamics). The controllability and observability matrices for the state-space model in Example 1 are given by:

$$\mathcal{C} = \begin{bmatrix} 0 & 0 & -1000 \\ 0 & -1000 & 50875 \\ 50 & -2500 & 125000 \end{bmatrix}, \qquad \mathcal{O} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & -.875 & -20 \\ 0 & -.875 & -20 \\ 0 & .7656 & 1017.5 \end{bmatrix}.$$

Both matrices have rank 3 so the system is both controllable and observable. □

1.5 Stability

The system (1.2) is *asymptotically stable* when all eigenvalues of A have negative real parts. In this case, for any bounded input $u(t)$ the output $y(t)$ and the state $x(t)$ are also bounded, i.e.,

Matlab hint 10. `eig(A)` computes the eigenvalues of the matrix `A`...

$$\|u(t)\| \leq c_1,\ \forall t \geq 0 \quad \Rightarrow \quad \|y(t)\| \leq c_2,\ \ \|x(t)\| \leq c_3 \quad \forall t \geq 0.$$

Moreover, if $u(t)$ converges to zero as $t \to \infty$, then $x(t)$ and $y(t)$ also converge to zero as $t \to \infty$.

Example 4 (Aircraft roll-dynamics). The eigenvalues of the matrix the A matrix for the state-space model in Example 1 are $\{0, -.875, -50\}$ so the system is not asymptotically stable. □

1.6 MATLAB hints

Matlab Hint 1 (`ss`). The command `sys_ss=ss(A,B,C,D)` assigns to `sys_ss` a MATLAB LTI state-space model with realization

$$\dot{x} = \mathtt{A}x + \mathtt{B}u, \qquad\qquad y = \mathtt{C}x + \mathtt{D}u.$$

Optionally, one can specify the names of the inputs, outputs, and state to be used in subsequent plots as follows:

```
sys_ss=ss(A,B,C,D,...
'InputName',{'input1','input2',...},...
'OutputName',{'output1','output2',...},...
'StateName',{'input1','input2',...})
```

The number of elements in the bracketed lists must match the number of inputs,outputs, and state variables. □

Matlab Hint 2 (`tf`). The command `sys _tf=tf(num,den)` assigns to `sys _tf` a MATLAB transfer-function. `num` is a vector with the coefficients of the numerator of the system's transfer-function and `den` a vector with the coefficients of the denominator. The last coefficient must always be the zero-order one. E.g., to get $\frac{2s}{s^2+3}$ you should use `num=[2 0];den=[1 0 3];`

For transfer-matrices, `num` and `den` are cell arrays. Type `help tf` for examples.

Optionally, one can specify the names of the inputs, outputs, and state to be used in subsequent plots as follows:

```
sys_tf=tf(num,den,...
'InputName',{'input1','input2',...},...
'OutputName',{'output1','output2',...},...
'StateName',{'input1','input2',...})
```

The number of elements in the bracketed lists must match the number of inputs,outputs, and state variables. □

Matlab Hint 3 (zpk). The command `sys_tf=zpk(z,p,k)` assigns to `sys_tf` a MATLAB transfer-function. `z` is a vector with the zeros of the system, `p` a vector with its poles, and `k` the gain. E.g., to get $\frac{2s}{(s+1)(s+3)}$ you should use `z=0;p=[1,3];k=2;`

For transfer matrices, `z` and `p` are cell arrays and `k` a regular array. Type `help zpk` for examples.

Optionally, one can specify the names of the inputs, outputs, and state to be used in subsequent plots as follows:

```
sys_tf=zpk(z,p,k,...
'InputName',{'input1','input2',...},...
'OutputName',{'output1','output2',...},...
'StateName',{'input1','input2',...})
```

The number of elements in the bracketed lists must match the number of inputs,outputs, and state variables. □

Matlab Hint 4 (tf). The commands `tf(sys_ss)` and `zpk(sys_ss)` compute the transfer-function of the state-space model `sys_ss` specified as in Matlab Hint 1.
`tf(sys_ss)` stores (and displays) the transfer function as a ratio of polynomials on s.
`zpk(sys_ss)` stores (and displays) the polynomials factored as the product of monomials (for the real roots) and binomials (for the complex roots). This form highlights the zeros and poles of the system. □

Matlab Hint 6 (ss). The command `ss(sys_tf)` computes the state-space model of the transfer function `sys` specified as in Matlab Hints 2 or 3. □

Matlab Hint 5 (expm). The command `expm(M)` computes the matrix exponential $e^{\mathtt{M}}$. With the symbolic toolbox, this command can be used to compute e^{At} symbolically as follows:

```
syms t
expm(A*t)
```

The first command defines `t` as a symbolic variable and the second computes $e^{\mathtt{At}}$ (assuming that the matrix `A` has been previously defined). □

Matlab Hint 7 (ctrb). The command `ctrb(sys)` computes the controllability matrix of the system `sys`. The system must be specified by a state-space model using, e.g., `sys=ss(A,B,C,D)`, where `A,B,C,D` are a realization of the system. Alternatively, one can use directly `ctrb(A,B)`. □

Matlab Hint 9 (obsv). The command `obsv(sys)` computes the observability matrix of the system `sys`. The system must be specified by a state-space model using, e.g., `sys=ss(A,B,C,D)`, where `A,B,C,D` are a realization of the system. Alternatively, one can use directly `obsv(A,C)`. □

Matlab Hint 10 (eig). The command `eig(A)` computes the eigenvalues of the matrix `A`. Alternatively, `eig(sys)` computes the eigenvalues of the A matrix for a state-space system `sys` specified by `sys=ss(A,B,C,D)`, where `A,B,C,D` are a realization of the system. □

Lecture #2

Linear Quadratic Regulation (LQR)

Contents

2.1 Feedback configuration

Figure 2.1 shows the feedback configuration for the *Linear quadratic regulation (LQR) problem.*

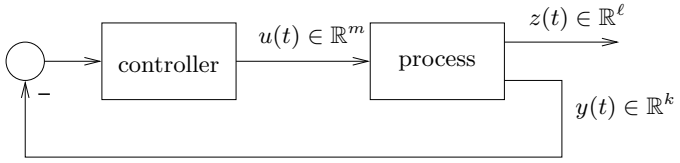


Figure 2.1. Linear quadratic regulation (LQR) feedback configuration. Note the *negative feedback* and the *absence of a reference signal.* Reference signals will be introduced in Lecture 4.

In this configuration, the state-space model of the process is of the form

$$\dot{x} = Ax + Bu, \qquad y = Cx, \qquad z = Gx + Hu. \tag{2.1}$$

and has two distinct outputs:

1. The *measured output* $y(t) \in \mathbb{R}^k$ corresponds to the signal(s) that can be measured and are therefore available for control. If the controller transfer-matrix is $C(s)$, we have
$$Y(s) = -C(s)U(s),$$
where $Y(s)$ and $U(s)$ denote the Laplace transforms of the process input $u(t)$ and the measured output $y(t)$, respectively.

2. The *controlled output* $z(t) \in \mathbb{R}^\ell$ corresponds to a signal that one would like to make as small as possible in the shortest possible amount of time.

   Sometimes $z(t) = y(t)$, which means that our control objective is to make the whole measured output very small. However, when the output $y(t)$ is a vector, often one simply needs to make one of the measured outputs $y_1(t)$ small. In this case, one chooses $z(t) = y_1(t)$.

   In some situations one chooses
$$z(t) = \begin{bmatrix} y_1(t) \\ \dot{y}_1(t) \end{bmatrix},$$
which means that we want to make both the measured output $y_1(t)$ and its derivative $\dot{y}_1(t)$ very small. Many other options are possible.

   The choice of z should be viewed as a design parameter. In Section 2.5 we will study the impact of this choice in the performance of the closed-loop.

2.2 Optimal Regulation

The LQR problem is defined as follows:

Problem 1 (Optimal LQR). Find the controller transfer-matrix $C(s)$ that makes the following criteria as small as possible
$$J_{\text{LQR}} := \int_0^\infty \|z(t)\|^2 + \rho \|u(t)\|^2 dt, \tag{2.2}$$
where ρ is a positive constant. □

Notation 1. Given an m-vector,
$$v = \begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix},$$
$\|v\|$ denotes the Euclidean norm of v, i.e.,
$$\|v\| = v'v = \Big(\sum_{i=1}^m v_i^2\Big)^{\frac{1}{2}}.$$

The term
$$\int_0^\infty \|z(t)\|^2 dt$$
corresponds to the *energy of the controlled output* and the term
$$\int_0^\infty \|u(t)\|^2 dt$$
to the *energy of the control signal*. In LQR one seeks a controller that minimizes both energies. However, decreasing the energy of the controlled output will require a large control signal and a small control signal will lead to large controlled outputs. The role of the constant ρ is to establish a trade-off between these conflicting goals:

1. When we chose ρ very large, the most effective way to decrease J_{LQR} is to use little control, at the expense of a large controlled output.

2. When we chose ρ very small, the most effective way to decrease J_{LQR} is to obtain a very small controlled output, even if this is achieved at the expense of a large controlled output.

Often the *optimal LQR problem* is defined more generally and consists of finding the controller transfer-matrix $C(s)$ that minimizes

$$J_{\mathrm{LQR}} := \int_0^\infty z(t)'Qz(t) + \rho u'(t)Ru(t)dt, \tag{2.3}$$

where Q is an $\ell \times \ell$ symmetric positive-definite matrix, R an $m \times m$ symmetric positive-definite matrix, and ρ a positive constant.

Sidebar 2. The most general form for the quadratic criteria is

$$\int_0^\infty x'\bar{Q}x + u'\bar{R}u + 2x'\bar{N}u\,dt$$

...

Sidebar 3. A symmetric $q \times q$ matrix M is *positive-definite* if $x'Mx > 0$, for every nonzero vector $x \in \mathbb{R}^q$ ...

Bryson's rule A first choice for the matrices Q and R in (2.3) is given by the *Bryson's rule* [1, p. 537]: select Q and R diagonal with

$$Q_{ii} = \frac{1}{\text{maximum acceptable value of } z_i^2}, \qquad i \in \{1, 2, \ldots, \ell\}$$
$$R_{jj} = \frac{1}{\text{maximum acceptable value of } u_j^2}, \qquad j \in \{1, 2, \ldots, m\},$$

which corresponds to the following criteria

$$J_{\mathrm{LQR}} := \int_0^\infty \Big(\sum_{i=1}^{\ell} Q_{ii}\, z_i(t)^2 + \rho \sum_{j=1}^{m} R_{jj}\, u_j(t)^2 \Big) dt.$$

In essence the Bryson's rule scales the variables that appear in J_{LQR} so that the *maximum acceptable value for each term is one*. This is especially important when the units used for the different components of u and z make the values for these variables numerically very different from each other.

Although Bryson's rule sometimes gives good results, often it is just the starting point to a trial-and-error iterative design procedure aimed at obtaining desirable properties for the closed-loop system. In Section 2.5 we will discuss systematic methods to chose the weights in the LQR criterion.

2.3 State-Feedback LQR

In the *state-feedback* version of the LQR problem (Figure 2.2), we assume that the whole state x can be measured and therefore it is available for control.

Solution to the optimal state-feedback LQR Problem 1. The optimal state-feedback LQR con-

Matlab hint 11. `lqr` computes the optimal state-feedback controller gain K...

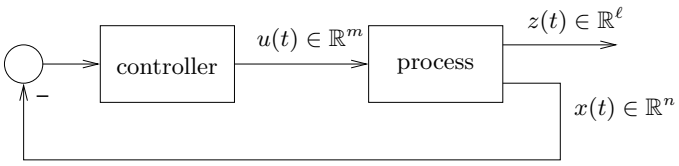


Figure 2.2. Linear quadratic regulation (LQR) with state feedback

troller for the criteria (2.3) is a simple matrix gain of the form

$$u = -Kx \tag{2.4}$$

where K is the $m \times n$ matrix given by

$$K = (H'QH + \rho R)^{-1}(B'P + H'QG)$$

and P is the unique positive-definite solution to the following equation

$$A'P + PA + G'QG - (PB + G'QH)(H'QH + \rho R)^{-1}(B'P + H'QG) = 0,$$

known as the *Algebraic Riccati Equation (ARE).* □

2.4 Stability and Robustness

Sidebar 4. A system $\dot{x} = Ax + Bu$ is *asymptotically stable* when all eigenvalues of A have negative real parts. See Section 1.5.

The state-feedback control law (2.4), results in a closed-loop system of the form

$$\dot{x} = (A - BK)x.$$

A crucial property of LQR controller design is that this *closed-loop is asymptotically stable* (i.e., all the eigenvalues of $A - BK$ have negative real part) as long as the following two conditions hold:

Sidebar 5. The definitions and tests for controllability and observability are reviewed in Section 1.4.

1. The system (2.1) is controllable.
2. The system (2.1) is observable when we ignore y and regard z as the sole output .

Attention! When selecting the measured output z, it is important to verify that the observability condition is satisfied.

Perhaps even more important is the fact that LQR controllers are *inherently robust with respect to process uncertainty.* To understand why, consider the open-loop transfer-matrix from the process' input u to the controller's output $\bar{u}$ (Figure 2.3). The state-space model

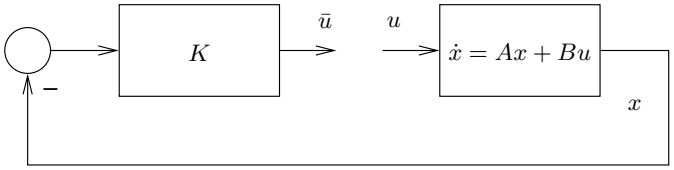


Figure 2.3. State-feedback open-loop gain

from u to $\bar{u}$ is given by

$$\dot{x} = Ax + Bu, \qquad \bar{u} = -Kx,$$

which corresponds to the following open-loop *negative* feedback $m \times m$ transfer-matrix

$$G_0(s) = K(sI - A)^{-1}B.$$

We focus our attention in single-input processes ($m = 1$), for which $G_0(s)$ is a scalar transfer-function and the following holds:

Kalman's Inequality. *When $H'G = 0$, the Nyquist plot of $G_0(j\omega)$ does not enter a circle of radius one around -1, i.e.,*

$$|1 + G_0(j\omega)| \geq 1, \qquad \forall \omega \in \mathbb{R}. \qquad \square$$

Sidebar 6. LQR controllers also exhibit robustness properties for multiple-input processes. However, in this case $G_0(s)$ is a $m \times m$ transfer-matrix and one needs a *multi-variable Nyquist criterion.*

Kalman's Inequality is represented graphically in Figure 2.4 and has several significant implications, which are discussed next.

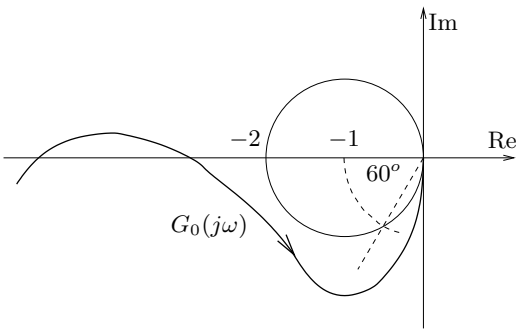


Figure 2.4. Nyquist plot for a LQR state-feedback controller

Positive gain margin If the process' gain is multiplied by a constant $k > 1$, its Nyquist plot simply expands radially and therefore the number of encirclements does not change. This corresponds to a *positive gain margin of* $+\infty$.

Negative gain margin If the process' gain is multiplied by a constant $.5 < k < 1$, its Nyquist plot contracts radially but the number of encirclements still does not change. This corresponds to a *negative gain margin of* $20 \log_{10}(.5) = -6dB$.

Phase margin If the process' phase increases by $\theta \in [-60, 60]$ degrees, its Nyquist plots rotates by θ but the number of encirclements still does not change. This corresponds to a *phase margin of* ± 60 *degrees.*

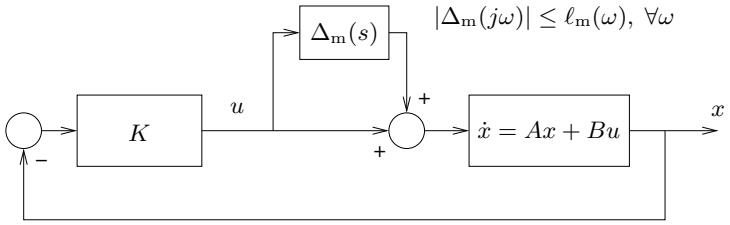


Figure 2.5. Unity feedback configuration with multiplicative uncertainty

Sidebar 7. Why?...

Multiplicative uncertainty Kalman's inequality guarantees that

$$\left|\frac{G_0(j\omega)}{1+G_0(j\omega)}\right| \le 2. \tag{2.5}$$

Since, we known that the closed-loop system in Figure 2.5 remains stable for every multiplicative uncertainty block $\Delta_{\rm m}(j\omega)$ with norm smaller than $\ell_{\rm m}(\omega)$, as long as

$$\left|\frac{G_0(j\omega)}{1+G_0(j\omega)}\right| < \frac{1}{\ell_{\rm m}(\omega)}, \tag{2.6}$$

we conclude that an LQR controller provides *robust stability with respect to any multiplicative uncertainty with magnitude smaller than* $\frac{1}{2}$, because we then have

$$\left|\frac{G_0(j\omega)}{1+G_0(j\omega)}\right| \le 2 < \frac{1}{\ell_{\rm m}(\omega)}.$$

However, much larger additive uncertainties may be admissible: e.g., when $G_0(j\omega) \gg 1$, (2.6) will hold for $\ell_{\rm m}(\omega)$ almost equal to 1; and when $G_0(j\omega) \ll 1$, (2.6) will hold for $\ell_{\rm m}(\omega)$ almost equal to $1/G_0(j\omega) \gg 1$.

Attention! Kalman's inequality is only valid when $H'G = 0$. When this is not the case, LQR controllers can be significantly less robust. This limits to same extent the controlled outputs that can be placed in z.

For example, consider the process $\dot{x} = Ax + Bu$ and suppose that we want to regulate a particular output $y_1 = C_1 x$. Choosing

$$z = y_1 = C_1 x$$

leads to $G = C_1$ and $H = 0$ and therefore $H'G = 0$, for which Kalman's inequality holds. However, choosing

$$z = \begin{bmatrix} y_1 \\ \dot{y}_1 \end{bmatrix} = \begin{bmatrix} C_1 x \\ C_1 \dot{x} \end{bmatrix} = \begin{bmatrix} C_1 x \\ C_1 Ax + C_1 Bu \end{bmatrix} = \begin{bmatrix} C_1 \\ C_1 A \end{bmatrix} x + \begin{bmatrix} 0 \\ C_1 B \end{bmatrix} u,$$

Sidebar 8. If the transfer function from u to y_1 has two more poles than zeros, then one can show that $C_1 B = 0$ and $H = 0$. In this case, Kalman's inequality holds also for this choice of z.

leads to

$$G = \begin{bmatrix} C_1 \\ C_1 A \end{bmatrix}, \qquad H = \begin{bmatrix} 0 \\ C_1 B \end{bmatrix},$$

and therefore

$$H'G = B'C_1'C_1A,$$

which may not be equal to zero. □

2.5 Loop-shaping control using LQR

Although Bryson's rule sometimes gives good results, it may not suffice to satisfy tight control specifications. We will see next a few other rules that allow us to actually do loop-shaping using LQR. We restrict our attention to the single-input case ($m = 1$) and $R = 1$, $Q = I$, which corresponds to

$$J_{\text{LQR}} := \int_0^\infty \|z(t)\|^2 + \rho\, u(t)^2 dt.$$

Low-frequency open-loop gain For the range of frequencies for which $|G_0(j\omega)| \gg 1$ (typically low frequencies), we have that

$$|G_0(j\omega)| \approx \frac{\|P_z(j\omega)\|}{\sqrt{H'H + \rho}}$$

Matlab hint 12. `sigma(sys)` draws the norm-Bode plot of the system `sys`...

where

$$P_z(s) := G(sI - A)^{-1}B + H$$

is the transfer function from the control signal u to the controlled output z. To understand the implications of this formula, it is instructive to consider two fairly typical cases:

1. When $z = y_1$, with $y_1 := C_1x$ scalar, we have

   $$|G_0(j\omega)| \approx \frac{|P_1(j\omega)|}{\sqrt{H'H + \rho}}.$$

   where

   $$P_1(s) := C_1(sI - A)^{-1}B$$

   is the transfer function from the control input u to the output y_1. In this case,

   (a) the "shape" of the magnitude of the open-loop gain $G_0(j\omega)$ is determined by the magnitude of the transfer function from the control input u to the output y_1;

   (b) the parameter ρ moves the magnitude Bode plot up and down (more precisely $H'H + \rho$).

   Sidebar 9. Although the magnitude of $G_0(j\omega)$ mimics the magnitude of $P_1(j\omega)$, the phase of the open-loop gain $G_0(j\omega)$ always leads to a stable closed-loop with an appropriate phase margin.

2. When $z = \begin{bmatrix} y_1 & \gamma\dot{y}_1 \end{bmatrix}'$, we can show that

   $$|G_0(j\omega)| \approx \frac{|1 + j\gamma\omega|\,|P_1(j\omega)|}{\sqrt{H'H + \rho}}. \tag{2.7}$$

   Sidebar 10. Why?...

   In this case the low-frequency open-loop gain mimics the process transfer function from u to y, with an extra zero at $1/\gamma$ and scaled by $\frac{1}{\sqrt{H'H+\rho}}$. Thus

(a) ρ moves the magnitude Bode plot up and down (more precisely $H'H + \rho$),

(b) large values for γ lead to a low-frequency zero and generally result in a larger phase margin (above the minimum of 60 degrees) and smaller overshoot in the step response. However, this is often achieved at the expense of a slower response.

High-frequency open-loop gain For $\omega \gg 1$, we have that

$$|G_0(j\omega)| \approx \frac{c}{\omega\sqrt{\rho}},$$

for some constant c. We thus conclude the following:

1. LQR controllers always exhibit a high-frequency magnitude decay of -20dB/decade.
2. The cross-over frequency is approximately given by

$$\frac{c}{\omega_{\text{cross}}\sqrt{\rho}} \approx 1 \quad \Leftrightarrow \quad \omega_{\text{cross}} \approx \frac{c}{\sqrt{\rho}},$$

which shows that the cross-over frequency is proportional to $1/\sqrt{\rho}$ and generally small values for ρ result in faster step responses.

Attention! The (slow) -20dB/decade magnitude decrease is the main shortcoming of state-feedback LQR controllers because it may not be sufficient to clear high-frequency upper bounds on the open-loop gain needed to reject disturbances and/or for robustness with respect to process uncertainty. We will see in Section 3.6 that this can actually be improved with output-feedback controllers. □

Example 5 (Aircraft roll-dynamics). Figure 2.6 shows Bode plots of the open-loop gain $G_0(s) = K(sI - A)^{-1}B$ for several LQR controllers obtained for the aircraft roll-dynamics in Example 1. The controlled output was chosen to be $z := \begin{bmatrix}\theta & \gamma\dot{\theta}\end{bmatrix}'$, which corresponds to

$$G := \begin{bmatrix}1 & 0 & 0\\ 0 & \gamma & 0\end{bmatrix}, \qquad H := \begin{bmatrix}0\\ 0\end{bmatrix}.$$

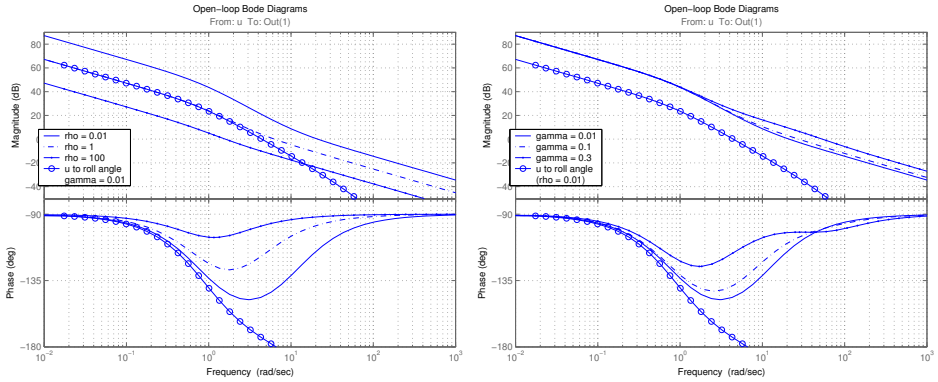
The controllers were obtained with $R = 1$, $Q = I_{2\times 2}$, and several values for ρ and γ. Figure 2.6(a) shows the open-loop gain for several values of ρ and Figure 2.6(b) shows the open-loop gain for several values of γ.

Figure 2.7 shows Nyquist plots of the open-loop gain $G_0(s) = K(sI-A)^{-1}B$ for different choices of the controlled output z. In Figure 2.7(a) $z := \begin{bmatrix}\theta & \dot{\theta}\end{bmatrix}'$, which corresponds to

$$G := \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}, \qquad H := \begin{bmatrix}0\\ 0\end{bmatrix}.$$

In this case, $H'G = \begin{bmatrix}0 & 0 & 0\end{bmatrix}$ and Kalman's inequality holds as can be seen in the Nyquist plot. In Figure 2.7(b), the controlled output was chosen to be $z := \begin{bmatrix}\theta & \dot{\tau}\end{bmatrix}'$, which corresponds to

$$G := \begin{bmatrix}1 & 0 & 0\\ 0 & 0 & -50\end{bmatrix}, \qquad H := \begin{bmatrix}0\\ 50\end{bmatrix}.$$



(a) Open-loop gain for several values of ρ. This parameter allow us to move the whole magnitude Bode plot up and down.

(b) Open-loop gain for several values of γ. Larger values for this parameter result in a larger phase margin.

Figure 2.6. Bode plots for the open-loop gain of the LQR controllers in Example 5. As expected, for low frequencies the open-loop gain magnitude matches that of the process transfer function from u to θ (but with significantly lower/better phase) and at high-frequencies the gain's magnitude falls at -20dB/decade.

In this case we have $H'G = \begin{bmatrix} 0 & 0 & -2500 \end{bmatrix}$ and Kalman's inequality does not hold. We can see from the Nyquist plot that the phase and gain margins are very small and there is little robustness with respect to unmodeled dynamics since a small perturbation in the process can lead to an encirclement of the point -1. □

2.6 MATLAB hints

Matlab Hint 11 (`lqr`). The command `[K,S,E]=lqr(A,B,QQ,RR,NN)` computes the optimal state-feedback LQR controller for the process

$$\dot{x} = \mathtt{A}x + \mathtt{B}u$$

with criteria

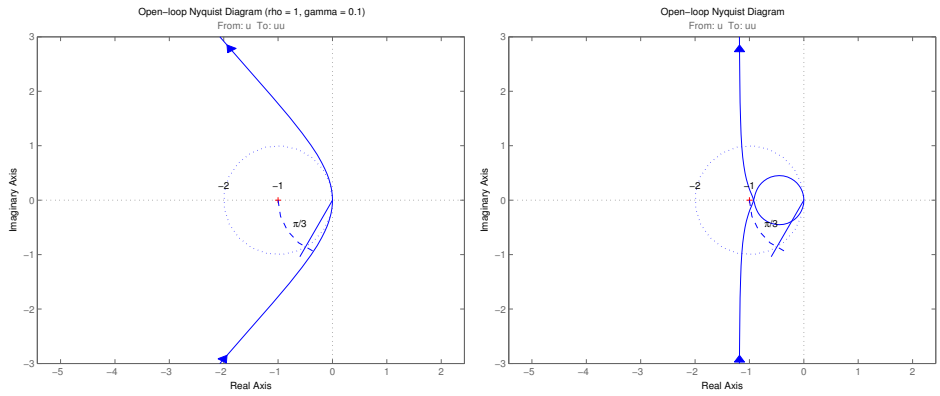
$$J := \int_0^\infty x(t)'\mathtt{QQ}x(t) + u'(t)\mathtt{RR}u(t) + 2x'(t)\mathtt{NN}u(t)dt.$$

For the criterion in (2.2) one should select

$$\mathtt{QQ} = G'G, \qquad \mathtt{RR} = H'H + \rho I, \qquad \mathtt{NN} = G'H$$

and for the criterion (2.3)

$$\mathtt{QQ} = G'QG, \qquad \mathtt{RR} = H'QH + \rho R, \qquad \mathtt{NN} = G'QH.$$



(a) $H'G = 0$ (b) $H'G \neq 0$

Figure 2.7. Bode plots for the open-loop gain of the LQR controllers in Example 5

(cf. Sidebar 2). This command returns the optimal state-feedback matrix `K`, the solution `P` to the corresponding Algebraic Riccati Equation, and the poles `E` of the closed-loop system. □

Matlab Hint 12 (`sigma`). The command `sigma(sys)` draws the norm-Bode plot of the system `sys`. For scalar transfer functions this command plots the usual magnitude Bode plot but for vector transfer function it plots the norm of the transfer function versus the frequency. □

2.7 To probe further

Sidebar 2 (General LQR). The most general form for a quadratic criteria is

$$J := \int_0^\infty x(t)'\bar{Q}x(t) + u'(t)\bar{R}u(t) + 2x'(t)\bar{N}u(t)dt. \tag{2.8}$$

Since $z = Gx + Hu$, the criterion in (2.2) is a special form of (2.8) with

$$\bar{Q} = G'G, \qquad \bar{R} = H'H + \rho I, \qquad \bar{N} = G'H$$

and (2.3) is a special form of (2.8) with

$$\bar{Q} = G'QG, \qquad \bar{R} = H'QH + \rho R, \qquad \bar{N} = G'QH.$$

For this criteria, the optimal state-feedback LQR controller is still of the form

$$u = -\bar{K}x$$

but now $\bar{K}$ is given by

$$\bar{K} = \bar{R}^{-1}(B'\bar{P} + \bar{N})$$

and $\bar{P}$ is a solution to the following *Algebraic Riccati Equation (ARE)*

$$A'P + PA + \bar{Q} - (\bar{P}B + \bar{N})\bar{R}^{-1}(B'\bar{P} + \bar{N}') = 0. \qquad \square$$

Sidebar 3. Other choices for the matrices Q and R are possible but one should always choose both matrices to be positive-definite. We recall that a symmetric $q \times q$ matrix M is *positive-definite* if

$$x'Mx > 0,$$

for every nonzero vector $x \in \mathbb{R}^q$. To test if a matrix is positive define one can compute its eigenvalues. If they are all positive the matrix is positive-definite, otherwise it is not (cf. Matlab Hint 10). $\square$

Sidebar 7 (Multiplicative uncertainty). Since the Nyquist plot of $G_0(j\omega)$ does not enter a circle of radius one around -1, we have that

$$|1 + G_0(j\omega)| \geq 1 \quad \Rightarrow \quad \left|\frac{1}{1 + G_0(j\omega)}\right| = \left|1 - \frac{G_0(j\omega)}{1 + G_0(j\omega)}\right| \leq 1 \quad \Rightarrow \quad \left|\frac{G_0(j\omega)}{1 + G_0(j\omega)}\right| \leq 2.$$

$\square$

Sidebar 10 (Equation (2.7)**).** When $z = \begin{bmatrix} y & \gamma\dot{y} \end{bmatrix}'$, we have that

$$z = \begin{bmatrix} y \\ \gamma\dot{y} \end{bmatrix} = \begin{bmatrix} Cx \\ \gamma CAx + \gamma CBu \end{bmatrix} \quad \Rightarrow \quad G = \begin{bmatrix} C \\ \gamma CA \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ \gamma CB \end{bmatrix}.$$

In this case,

$$P_z(s) = \begin{bmatrix} P_y(s) \\ \gamma s P_y(s) \end{bmatrix} = \begin{bmatrix} 1 \\ \gamma s \end{bmatrix} P_y(s),$$

where $P_y(s) := C(sI - A)^{-1}B$, and therefore

$$|G_0(j\omega)| \approx \frac{\sqrt{1 + \gamma^2\omega^2}\,|P_y(j\omega)|}{\sqrt{H'H + \rho}} = \frac{|1 + j\gamma\omega|\,|P_y(j\omega)|}{\sqrt{H'H + \rho}}. \qquad \square$$

2.8 Exercises

Exercise 1. Verify using the diagram in Figure 4.1 that, for the single-input case ($m = 1$), the closed-loop transfer function $T_u(s)$ from the reference r to the process input u is given by

$$T_u(s) = \frac{1}{1 + G_0}(KF + N),$$

where $G_0(s) = K(sI - A)^{-1}B$, and the closed-loop transfer function T_z from the reference r to the controlled output z is given by

$$T_z(s) = \frac{1}{1 + G_0} P_z(KF + N),$$

where $P_z(s) = G(sI - A)^{-1}B + H$. □

Exercise 2. Consider an inverted pendulum operating near the upright equilibrium position, with linearized model given by

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{g}{\ell} & -\frac{b}{m\ell^2} \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix} T$$

where T denotes an applied torque, θ the pendulum's angle with a vertical axis pointing up, and $\ell = 1\text{m}$, $m = 1\text{Kg}$, $b = .1\text{N/m/s}$, $g = 9.8\text{m/s}^2$.

1. Design a PD controller using LQR
2. Design a PID controller using LQR

   Hint: Consider an "augmented" process model with state θ, $\dot{\theta}$, $z(t) = \int_0^t \theta(s)ds$.

Lecture #3

LQG/LQR Output Feedback

Contents

3.1 Output Feedback

The state-feedback LQR formulation considered in Chapter 2.3 suffered from the drawback that the optimal control law

$$u(t) = -Kx(t) \tag{3.1}$$

required the whole state x of the process to be measurable. An possible approach to overcome this difficulty is to estimate the state of the process based solely on the measured output y, and use

$$u(t) = -K\hat{x}(t)$$

instead of (3.1), where $\hat{x}(t)$ denotes an estimate of the process' state $x(t)$. In this chapter we consider the problem of constructing state estimates.

3.2 Full-order observers

Consider a process with state-space model

$$\dot{x} = Ax + Bu, \qquad y = Cx, \tag{3.2}$$

where y denotes the measured output, u the control input. We assume that x cannot be measured and our goal to estimate its value based on y.

Suppose we construct the estimate $\hat{x}$ by replicating the process dynamics as in

$$\dot{\hat{x}} = A\hat{x} + Bu. \tag{3.3}$$

To see of this would generate a good estimate for x, we can define the state estimation error $e := x - \hat{x}$ and study its dynamics. From (3.2) and (3.3), we conclude that

$$\dot{e} = Ax - A\hat{x} = Ae.$$

This shows that when the matrix A is asymptotically stable the error e converges to zero *for any input* u, which is good news because it means that $\hat{x}$ eventually converges to x as $t \to \infty$. However, when A is not stable e is unbounded and $\hat{x}$ grow further and further apart from x as $t \to \infty$. To avoid this, one includes a correction term in (3.3):

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}), \qquad \hat{y} = C\hat{x}, \tag{3.4}$$

where $\hat{y}$ should be viewed as an estimate of y and L a given $n \times k$ matrix. When $\hat{x}$ is equal (or very close) to x, then $\hat{y}$ will be equal (or very close) to y and the correction term $L(y-\hat{y})$ plays no role. However, when $\hat{x}$ grows away from x, this term will (hopefully!) correct the error. To see how this can be done, we re-write the estimation error dynamics now for (3.2) and (3.4):

$$\dot{e} = Ax - A\hat{x} - L(Cx - C\hat{x}) = (A - LC)e.$$

Now e converges to zero as long as $A - LC$ is asymptotically stable, It turns out that, even when A is unstable, in general we will be able to select L so that $A - LC$ is asymptotically stable. The system (3.4) can be re-write as

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly, \tag{3.5}$$

Sidebar 11. "Full-order" refers to the fact that the size of its state $\hat{x}$ is equal to the size of the process' state x.

and is called a *full-order observer* for the process. Full-order observers have two inputs—the process' control input u and its measured output y—and a single output—the state estimate $\hat{x}$. Figure 3.1 shows how a full-order observer is connected to the process.

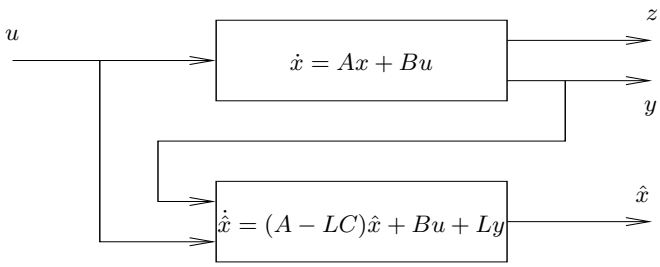


Figure 3.1. Full-order observer

3.3 LQG estimation

Any choice of L in (3.4) for which $A - LC$ is asymptotically stable will make $\hat{x}$ converge to x, as long at the process dynamics is given by (3.2). However, in general the output y is affected by measurement noise and the process dynamics are also affected by disturbance. In light of this, a more reasonable model for the process is

$$\dot{x} = Ax + Bu + \bar{B}d, \qquad y = Cx + n, \tag{3.6}$$

where d denotes a disturbance and n measurement noise. In this we need to re-write the estimation error dynamics for (3.6) and (3.4), which leads to

$$\dot{e} = Ax + \bar{B}d - A\hat{x} - L(Cx + n - C\hat{x}) = (A - LC)e + \bar{B}d - Ln.$$

Because of n and d, the estimation error will generally not converge to zero, but we would still like it to remain small by appropriate choice of the matrix L. This motivates the so called *Linear-quadratic Gaussian (LQG) estimation problem*:

Problem 2 (Optimal LQG). Find the matrix gain L that minimizes the asymptotic expected value of the estimation error:

$$J_{\text{LQG}} := \lim_{t\to\infty} \mathrm{E}\left[\|e(t)\|^2\right],$$

where $d(t)$ and $n(t)$ are zero-mean Gaussian noise processes (uncorrelated from each other) with power spectrum

$$S_d(\omega) = Q_N, \qquad S_n(\omega) = R_N, \qquad \forall\omega. \qquad \square$$

Solution to the optimal LQG Problem 2. The optimal LQG estimator gain L is the $n \times k$ matrix given by

$$L = PC'R_N^{-1}$$

and P is the unique positive-definite solution to the following *Algebraic Riccati Equation (ARE)*

$$AP + PA' + \bar{B}Q_N\bar{B}' - PC'R_N^{-1}CP = 0.$$

When one uses the optimal gain L in (3.5), this system is called the *Kalman-Bucy* filter.

A crucial property of this system is that $A - LC$ is asymptotically stable as long as the following two conditions hold:

1. The system (3.6) is observable.
2. The system (3.6) is controllable when we ignore u and regard d as the sole input.

Different choices of Q_N and R_N result in different estimator gains L:

Sidebar 12. A zero-mean white noise process n has an autocorrelation of the form

$$R_n(t_1, t_2) := \mathrm{E}\left[n(t_1)\, n'(t_2)\right] = Q_N\, \delta(t_1 - t_2).$$

Such process is *wide-sense stationary* in the sense that its mean is time-invariant and its autocorrelation $R(t_1, t_2)$ only depends on the difference $\tau := t_1 - t_2$. Its power spectrum is frequency-independent and given by

$$S_n(\omega) := \int_{-\infty}^{\infty} R(\tau) e^{-j\omega\tau} d\tau = Q_N.$$

Matlab hint 13. `kalman` computes the optimal LQG estimator gain L...

Sidebar 13. A symmetric $q \times q$ matrix M is *positive definite* if $x'Mx > 0$, for every nonzero vector $x \in \mathbb{R}^q$ (cf. Sidebar 3),

1. When R_N is *very small* (when compared to Q_N), the measurement noise n is necessarily small so the optimal estimator interprets a large deviation of $\hat{y}$ from y as an indication that the estimate $\hat{x}$ is bad and needs to be correct. In practice, this lead to large matrices L and fast poles for $A - LC$.

2. When R_N is *very large*, the measurement noise n is large so the optimal estimator is much more conservative in reacting to deviations of $\hat{y}$ from y. This generally leads to smaller matrices L and slow poles for $A - LC$.

We will return to the selection of Q_N and R_N in Section 3.6.

3.4 LQG/LQR output feedback

We now go back to the problem of designing an output-feedback controller for the process:

$$\dot{x} = Ax + Bu, \qquad y = Cx, \qquad z = Gx + Hu.$$

Suppose that we designed a state-feedback controller

$$u = -Kx \tag{3.7}$$

that solves an LQR problem and constructed an LQG state-estimator

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly.$$

Matlab hint 14. `reg(sys,K,L)` computes the LQG/LQR *positive* output-feedback controller for the process `sys` with regulator gain `K` and estimator gain `L`...

We can obtain an output-feedback controller by using the estimated state $\hat{x}$ in (3.7), instead of the true state x. This leads to the following output-feedback controller

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly = (A - LC - BK)\hat{x} + Ly, \qquad u = -K\hat{x},$$

with *negative-feedback* transfer matrix given by

$$C(s) = K(sI - A + LC + BK)^{-1}L.$$

This is usually known as an *LQG/LQR output-feedback controller* and the resulting closed-loop is shown in Figure 3.2.

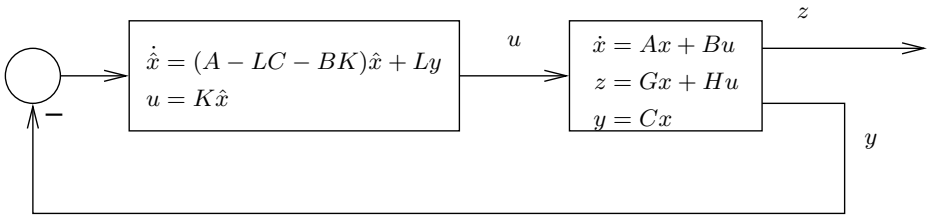


Figure 3.2. LQG/LQR output-feedback

3.5 Separation Principle

The first question to ask about an LQG/LQR controller is whether or not the closed-loop system will be stable. To answer this question we collect all the equations that defines the closed-loop system:

$$\dot{x} = Ax + Bu, \qquad y = Cx, \tag{3.8}$$

$$\dot{\hat{x}} = (A - LC)\hat{x} + Bu + Ly, \qquad u = -K\hat{x}. \tag{3.9}$$

To check the stability of this system it is more convenient to consider the dynamics of the estimation error $e := x - \hat{x}$ instead of the the state estimate $\hat{x}$. To this effect we replace in the above equations $\hat{x}$ by $x - e$, which yields:

$$\begin{aligned} \dot{x} &= Ax + Bu = (A - BK)x + BKe, & y &= Cx, \\ \dot{e} &= (A - LC)e, & u &= -K(x - e). \end{aligned}$$

This can be written in matrix notation as

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}, \qquad y = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix}.$$

Separation Principle. *The eigenvalues of the closed-loop system* (3.8) *are given by those of the state-feedback regulator dynamics $A - BK$ together with those of state-estimator dynamics $A - LC$. In case these both matrices are asymptotically stable, then so is the closed-loop* (3.8).

Sidebar 14. Any eigenvalue of a block diagonal matrix must be an eigenvalue of one of the diagonal blocks.

3.6 Loop-gain recovery

We saw in Sections 2.4 and 2.5 that state-feedback LQR controllers have desirable robustness properties and that we can shape the open-loop gain by appropriate choice of the LQR weighting parameter ρ and the choice of the controlled output z. It turns out that we can, to some extent, recover the LQR open-loop gain for the LQG/LQR controller.

Loop-gain recovery. *Suppose that the process is single-input/single-output and has* no zeros in the right half-place. *Selecting*

$$\bar{B} := B, \qquad R_N := \sigma, \quad \sigma > 0,$$

the open-loop gain for the output-feedback LQG/LQR controller converges to the open-loop gain for the state-feedback LQR state-feedback controller over a range of frequencies $[0, \omega_{\max}]$ as we make $\sigma \to 0$, i.e.,

$$C(j\omega)P(j\omega) \quad \xrightarrow{\sigma \to 0} \quad K(j\omega I - 1)^{-1}B, \quad \forall \omega \in [0, \omega_{\max}]$$

In general, the larger $\omega_{\max}$ is, the smaller σ needs to be for the gains to match.

Sidebar 15. $\bar{B} = B$ corresponds to an *input disturbance* since the process becomes

$$\begin{aligned} \dot{x} &= Ax + Bu + \bar{B}d \\ &= Ax + B(u + d). \end{aligned}$$

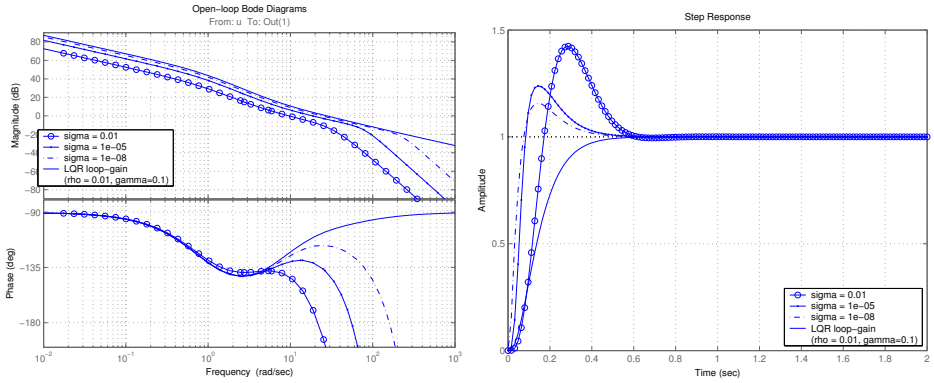
Attention! 1. To achieve loop-gain recovery we need to chose $R_N := \sigma$, *even if this does not accurately describe the noise statistics.* This means that the estimator may not be optimal for the actual noise.

2. One should not make σ smaller than necessary because we do not want to recover the (slow) -20dB/decade magnitude decrease at high frequencies. In practice we should make σ *just small enough to get loop-recovery until just above or at cross-over.* For larger values of ω, the output-feedback controller may actually behave much better than the state-feedback one.

3. When the process has *zeros in the right half-plane,* loop-gain recovery will generally only work up to the frequencies of the nonminimum-phase zeros.

   When the zeros are in the *left half-plane but close to the axis,* the closed-loop will not be very robust with respect to uncertainty in the position of the zeros. This is because the controller will attempt to cancel these zeros. □

Example 6 (Aircraft roll-dynamics). Figure 3.3(a) shows Bode plots of the open-loop gain for the state-feedback LQR state-feedback controller vs. the open-loop gain for several output-feedback LQG/LQR controller obtained for the aircraft roll-dynamics in Example 1. The LQR controller was designed using the controlled output $z := \begin{bmatrix}\theta & \gamma\dot{\theta}\end{bmatrix}'$, $\gamma = .1$ and



(a) Open-loop gain (b) Closed-loop step response

Figure 3.3. Bode plots and closed-loop step response for the open-loop gain of the LQR controllers in Examples 6, 8.

$\rho = .01$. For the LQG state-estimators we used $\bar{B} = B$ and $R_N = \sigma$ for several values of σ. We can see that, as σ decreases, the range of frequencies over which the open-loop gain of the output-feedback LQG/LQR controller matches that of the state-feedback LQR state-feedback increases. Moreover, at high frequencies the output-feedback controllers exhibit much faster (and better!) decays of the gain's magnitude. □

3.7 MATLAB hints

Matlab Hint 13 (`kalman`). The command `[est,L,P]=kalman(sys,QN,RN)` computes the optimal LQG estimator gain for the process

$$\dot{x} = \mathtt{A}x + \mathtt{B}u + \mathtt{BB}d, \qquad\qquad y = \mathtt{C}x + n,$$

where $d(t)$ and $n(t)$ are zero-mean Gaussian noise processes (uncorrelated from each other) with power spectrum

$$S_d(\omega) = \mathtt{QN}, \qquad\qquad S_n(\omega) = \mathtt{RN}, \qquad\qquad \forall \omega.$$

The system `sys` should be a state-space model defined by `sys=ss(A,[B BB],C,0)`. This command returns the optimal estimator gain `L`, the solution `P` to the corresponding Algebraic Riccati Equation, and a state-space model `est` for the estimator. The inputs to `est` are $[u; y]$ and its outputs are $[\hat{y}; \hat{x}]$. □

Matlab Hint 14 (`reg`). The command `reg(sys,K,L)` computes a state-space model for a *positive* output-feedback LQG/LQG controller for the process with state-space model `sys` with regulator gain `K` and estimator gain `L`. □

3.8 Exercises

Exercise 3. Consider an inverted pendulum on a cart operating near the upright equilibrium position, with linearized model given by

$$\begin{bmatrix} p \\ \dot{p} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -2.94 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 11.76 & 0 \end{bmatrix} \begin{bmatrix} p \\ \dot{p} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ .325 \\ 0 \\ -.3 \end{bmatrix} F$$

where F denotes a force applied to the cart, p the cart's horizontal position, and θ the pendulum's angle with a vertical axis pointing up.

1. Design an LQG/LQR output-feedback controller that uses only the angle θ and the position of the cart p.

2. Design an LQG/LQR output-feedback controller that uses the angle θ, the angular velocity $\dot{\theta}$, the position of the cart p, and its derivative using LQG/LQR (full-state feedback).

   Why use LQG when the state is accessible? □

Lecture #4

Set-point control

Contents

4.1 Nonzero equilibrium state and input

Often one does not want to make z as small as possible, but instead make it converge as fast as possible to a given constant *set-point value* r. This can be achieved by making the state x and the input u of the process (2.1) converge to values x^* and u^* for which

Sidebar 16. We will see shortly how to accomplish this.

$$Ax^* + Bu^* = 0, \qquad r = Gx^* + Hu^*. \tag{4.1}$$

The right equation makes sure that z will be equal to r when x and u reach x^* and u^*, respectively. The left-equation makes sure that when these values are reached, $\dot{x} = 0$ and therefore x will remain equal to x^*, i.e., x^* is an *equilibrium state*.

Given the desired set-point r for x, *computing* x^* *and* u^* is straightforward because (4.1) is a system of linear equations and in general the solution to these equations is of the form

Sidebar 17. When the process transfer-function has an integrator, one generally gets $u^* = 0$.

$$x^* = Fr, \qquad u^* = Nr. \tag{4.2}$$

For example, when the number of inputs to the process m is equal to the number of controlled outputs ℓ, we have

Sidebar 18. The matrix $\left[\begin{smallmatrix} A & B \\ G & H \end{smallmatrix}\right]$ is invertible unless the process' transfer-function from u to z has a zero at the origin. This derivative effect will always make z converge to zero when the input converges to a constant.

$$\begin{bmatrix} A & B \\ G & H \end{bmatrix} \begin{bmatrix} x^* \\ u^* \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} x^* \\ u^* \end{bmatrix} = \begin{bmatrix} A & B \\ G & H \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix}, \tag{4.3}$$

and F is an $n \times \ell$ matrix given by the top n rows and right-most ℓ columns of $\left[\begin{smallmatrix} A & B \\ G & H \end{smallmatrix}\right]^{-1}$ and N is an $m \times \ell$ matrix given by the bottom m rows and right-most ℓ columns of $\left[\begin{smallmatrix} A & B \\ G & H \end{smallmatrix}\right]^{-1}$.

Attention! When the number of process inputs m is larger than the number of controlled outputs ℓ we have an *over-actuated system* and the system of equations (4.1) will generally have multiple solutions. One of them is

$$\begin{bmatrix} x^* \\ u^* \end{bmatrix} = \begin{bmatrix} A & B \\ G & H \end{bmatrix}' \left(\begin{bmatrix} A & B \\ G & H \end{bmatrix} \begin{bmatrix} A & B \\ G & H \end{bmatrix}' \right)^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix}.$$

In this case, we can still express x^* and u^* as in (4.2).

When the number of process inputs m is smaller than the number of controlled outputs ℓ we have an *under-actuated system* and the system of equations (4.1) may not have a solution. In fact, a solution will only exists for some specific references r. However, when it does exist it we can still express x^* and u^* as in (4.2). □

4.2 State-feedback

Sidebar 19. Why?...

When one wants z to converge to a given *set-point value* r, the state-feedback controller should be

$$u = -K(x - x^*) + u^* = -Kx + (KF + N)r, \tag{4.4}$$

where x^* and u^* are given by (4.2) and K is the gain of the optimal regulation problem. The corresponding control architecture is shown in Figure 4.1. The state-space model for

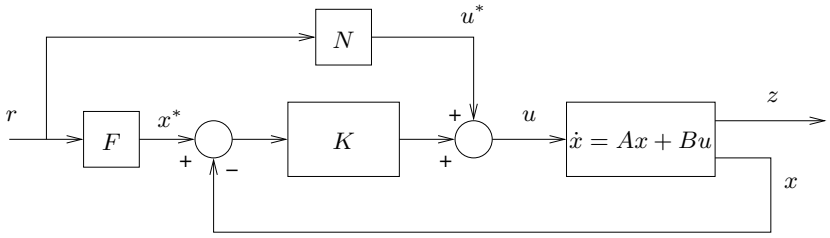
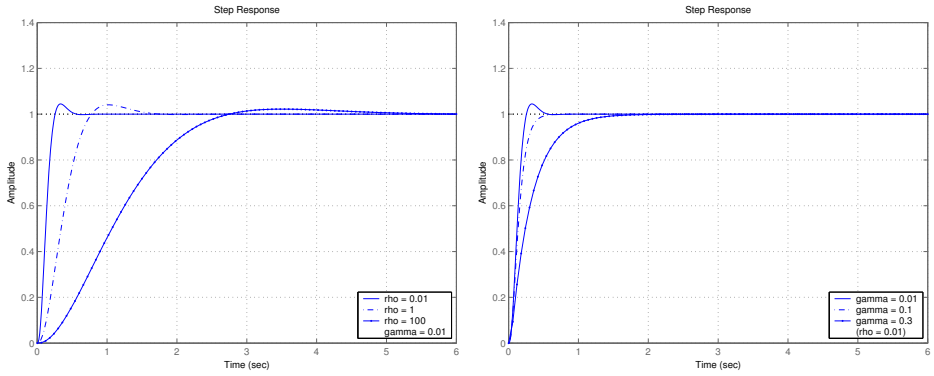


Figure 4.1. Linear quadratic set-point control with state feedback

the closed-loop system is given by

$$\begin{aligned} \dot{x} &= Ax + Bu = (A - BK)x + B(KF + N)r \\ z &= Gx + Hu = (G - HK)x + H(KF + N)r. \end{aligned}$$

Example 7 (Aircraft roll-dynamics). Figure 4.2 shows step responses for the state-feedback LQR controllers in Example 5, whose Bode plots for the open-loop gain are shown in Figure 2.6. Figure 4.2(a) shows that smaller values of ρ lead to faster responses and Figure 4.2(b) shows that larger values for γ lead to smaller overshoots (but slower responses). □



(a) (b)

Figure 4.2. Step responses for the closed-loop LQR controllers in Example 7

4.3 Output-feedback

When one wants z to converge to a given *set-point value* r, the output-feedback LQG/LQR controller should be

Sidebar 20. Why?...

$$\dot{\bar{x}} = (A - LC - BK)\bar{x} + L(Cx^* - y), \qquad u = K\bar{x} + u^*, \tag{4.5}$$

where x^* and u^* are given by (4.2). The corresponding control architecture is shown in Figure 4.3. The state-space model for the closed-loop system is given by

Sidebar 21. When $z = y$, we have $G = C$, $H = 0$ and in this case $Cx^* = r$. This corresponds to $CF = 1$ in Figure 4.3. When the process has an integrator we get $N = 0$ and obtain the usual unity-feedback configuration.

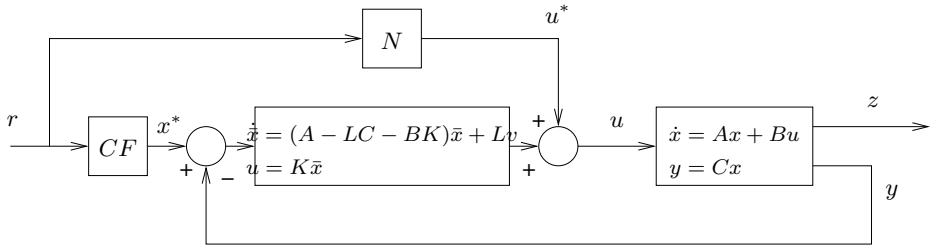


Figure 4.3. LQG/LQR set-point control

$$\begin{bmatrix} \dot{x} \\ \dot{\bar{x}} \end{bmatrix} = \begin{bmatrix} Ax + B(K\bar{x} + u^*) \\ (A - LC - BK)\bar{x} + L(Cx^* - Cx) \end{bmatrix} = \begin{bmatrix} A & BK \\ -LC & A - LC - BK \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \end{bmatrix} + \begin{bmatrix} BN \\ LCF \end{bmatrix} r$$

$$z = Gx + H(K\bar{x} + u^*) = \begin{bmatrix} G & HK \end{bmatrix} \begin{bmatrix} x \\ \bar{x} \end{bmatrix} + HNr$$

Example 8 (Aircraft roll-dynamics). Figure 3.3(b) shows step responses for the output-feedback LQG/LQR controllers in Example 6, whose Bode plots for the open-loop gain are

shown in Figure 3.3(a). We can see that smaller values of σ lead to a smaller overshoot mostly due to a larger gain margin. □

4.4 To probe further

Sidebar 19 (Set-point control with state-feedback). To understand why (4.4) works, suppose we define

$$\tilde{z} = z - r, \qquad \tilde{x} = x - x^*, \qquad \tilde{u} = u - u^*.$$

Then

$$\begin{aligned}\dot{\tilde{x}} &= Ax + Bu = A(x - x^*) + B(u - u^*) + Ax^* + Bu^* \\ \tilde{z} &= Gx + Hu - r = G(x - x^*) + H(u - u^*) + Gx^* + Hu^* - r\end{aligned}$$

and we conclude that

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}, \qquad \tilde{z} = G\tilde{x} + H\tilde{u}. \tag{4.6}$$

By selecting the control signal in (4.4), we are setting

$$\tilde{u} = u - u^* = -K(x - x^*) = -K\tilde{x},$$

which is the optimal state-feedback LQR controller that minimizes

$$J_{\text{LQR}} := \int_0^\infty \big(\tilde{z}(t)'Q\tilde{z}(t)\big) + \rho\big(\tilde{u}'(t)R\tilde{u}(t)\big)dt,$$

This controller makes the system (4.6) asymptotically stable and therefore $\tilde{x}$, $\tilde{u}$, $\tilde{z}$ all converge to zero as $t \to \infty$, which means that z converges to r. □

Sidebar 20 (Set-point control with output-feedback). To understand why (4.5) works suppose we define

$$\tilde{z} = z - r, \qquad \tilde{x} = x - x^* + \bar{x}.$$

Then

$$\dot{\tilde{x}} = (A - LC)\tilde{x} \tag{4.7}$$

$$\dot{\bar{x}} = (A - BK)\bar{x} - LC\tilde{x} \tag{4.8}$$

$$\tilde{z} = G(\tilde{x} - \bar{x}) + HK\bar{x} - r. \tag{4.9}$$

1. Since $A - LC$ is asymptotically stable, we conclude from (4.7) that $\tilde{x} \to 0$ as $t \to \infty$. In practice, we can view the state $\bar{x}$ of the controller as an estimate of $x^* - x$.
2. Since $A - BK$ is asymptotically stable and $\tilde{x} \to 0$ as $t \to \infty$, we conclude from (4.8) that $\bar{x} \to 0$ as $t \to \infty$.
3. Since $\tilde{x} \to 0$ and $\bar{x} \to 0$ as $t \to \infty$, we conclude from (4.9) that $z \to r$ as $t \to \infty$!

Exercise 4. Verify equations (4.7), (4.8), and (4.9). □

Bibliography

[1] G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.

[2] J. V. Vegte. *Feedback Control Systems*. Prentice Hall, New Jersey, 3rd edition, 1994.

[3] K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, New Jersey, 1996.