1

Introduction

1.1 Overview

The goal of this chapter is to provide a general overview of constrained control and estimation. This is intended to motivate the material to follow. Section 1.2 treats constrained control, Section 1.3 deals with constrained estimation, and Section 1.4 draws parallels between these two problems.

1.2 Introduction to Constrained Control

Handling constraints in control system design is an important issue in most, if not all, real world problems.

It is readily appreciated that all real world control systems have an associated set of constraints; for example, inputs always have maximum and minimum values and states are usually required to lie within certain ranges. Of course, one could proceed by ignoring these constraints and hope that no serious consequences result from this approach. This simple procedure may be sufficient at times. On the other hand, it is generally true that higher levels of performance are associated with operating on, or near, constraint boundaries. Thus, a designer really cannot ignore constraints without incurring a performance penalty.

As an illustration of these facts consider a simple automobile control problem. We mentioned in the Preface that there exist maximum and minimum available throttle displacements, that is, the system input is constrained. Other variables are also subject to constraints; for example, acceleration and deceleration have to be limited to prevent the vehicle's wheels from loosing traction. These factors constitute a constraint on the state of the system. Thus, modern cars incorporate both traction control (for acceleration) and anti-skid braking [ABS] (for deceleration). Both mechanisms ensure safe operation when variables are pushed to their limits.

As another simple example, consider the problem of rudder roll stabilisation of ships. The prime function of the rudder is to maintain the ship's heading. However, the rudder also imparts a rolling moment to the ship. Thus, the rudder can be used to achieve a measure of roll stabilisation. Since the rolling moment induced by the rudder is relatively small, it can be appreciated that large rudder displacements will be called upon, especially under heavy sea conditions. Of course, practical rudders must operate subject to constraints on both their total displacement (typically ± 30 degrees) and slew rate (typically ± 15 degrees per second). Indeed, it is generally agreed that rudder roll stabilisation can actually be counterproductive unless appropriate steps are taken to adequately deal with the presence of constraints. We will devote Chapter 14 to a more comprehensive introduction to rudder roll stabilisation. Other practical problems are discussed in Chapters 15 and 16.

Most of the existing literature on control theory deals with unconstrained problems. Nonetheless, as discussed above, there are strong practical reasons why a system should be operated on constraint boundaries. Thus, this book is aimed at going a step beyond traditional linear control theory to include consideration of constraints.

Our view of the existing methods for dealing with constraints in control system design is that they can be broadly classified under four headings:

- cautious
- serendipitous
- evolutionary
- tactical

In the "cautious" approach, one aims to explicitly deal with constraints by deliberately reducing the performance demands until the point where the constraints are not met at all. This has the advantage of allowing one to essentially use ordinary unconstrained design methods and hence to carry out a rigorous linear analysis of the problem. On the other hand, this is achieved at the cost of a potentially important loss in achievable performance since we expect high performance to be associated with pushing the boundaries, that is, acting on or near constraints.

In the "serendipitous" approach, one takes no special precautions to handle constraints, and hence occasional violation of the constraints is possible (that is, actuators reach saturation, states exceed their allowed values, and so on). Sometimes this can lead to perfectly acceptable results. However, it can also have a negative impact on important performance measures, including closed loop stability, since no special care is taken of the constrained phase of the response.

In the "evolutionary" approach, one begins with an unconstrained design philosophy but then adds modifications and embellishments to ensure that the negative consequences of constraints are avoided, or at least minimised, whilst ensuring that performance goals are attained. Examples of evolutionary

approaches include various forms of anti-windup control, high gain-low gain control, piecewise linear control and switching control.

One might suspect that, by careful design and appropriate use of intuition, one can obtain quite acceptable results from the evolutionary approach provided one does not push too hard. However, eventually, the constraints will override the usual linear design paradigm. Under these conditions, there could be advantages in "starting afresh". This is the philosophy of the so-called "tactical" approaches, in which one begins afresh with a formulation that incorporates constraints from the beginning in the design process. One way of achieving this is to set the problem up as a constrained optimisation problem. This will be the approach principally covered in this book.

Of course, the above classification does not cover all possibilities. Indeed, many methods fall into several categories.

To provide further motivation for this subject, we will present a simple example illustrating aspects of the cautious, serendipitous and tactical approaches.

We will base our design on linear quadratic regulator [LQR] theory. Thus, consider an objective function of the form:

$$V_N(\{x_k\},\{u_k\}) \triangleq \frac{1}{2}x_N^{\mathrm{T}}Px_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(x_k^{\mathrm{T}}Qx_k + u_k^{\mathrm{T}}Ru_k\right), \tag{1.1}$$

where $\{u_k\}$ denotes the *control sequence* $\{u_0, u_1, \ldots, u_{N-1}\}$, and $\{x_k\}$ denotes the corresponding *state sequence* $\{x_0, x_1, \ldots, x_N\}$. In (1.1), $\{u_k\}$ and $\{x_k\}$ are related by the linear state equation:

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, \ldots, N-1,$$

where x_0, the initial state, is assumed to be known.

In principle one can adjust the following parameters to obtain different manifestations of performance:

- the optimisation horizon N
- the state weighting matrix Q
- the control weighting matrix R
- the terminal state weighting matrix P

Actually, adjusting one or more of these parameters to manipulate key performance variables turns out to be one of the principal practical attributes of constrained linear control. We illustrate some of the basic features of constrained control using the objective function (1.1) via the following simple example.

Example 1.2.1. Consider the specific linear system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ y_k &= Cx_k, \end{aligned} \tag{1.2}$$

with

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix},$$

which is the zero-order hold discretisation with sampling period 1 of the double integrator

$$\frac{d^2y(t)}{dt^2} = u(t).$$

We take the initial condition (for illustrative purposes) to be $x_0 = \begin{bmatrix} -6 & 0 \end{bmatrix}^{\mathrm{T}}$ and suppose that the actuators have maximum and minimum values (saturation) so that the control magnitude is constrained such that $|u_k| \leq 1$ for all k. We will design cautious, serendipitous, and tactical feedback controllers for this system. A schematic of the feedback control loop is shown in Figure 1.1, where "sat" represents the actuator modelled by the *saturation function*

$$\mathrm{sat}(u) \triangleq \begin{cases} 1 & \text{if } u > 1, \\ u & \text{if } |u| \leq 1, \\ -1 & \text{if } u < -1. \end{cases} \tag{1.3}$$

Note that the section of Figure 1.1 in the dashed-line box is part of the physical reality and is not subject to change (unless, of course, the actuator is replaced).

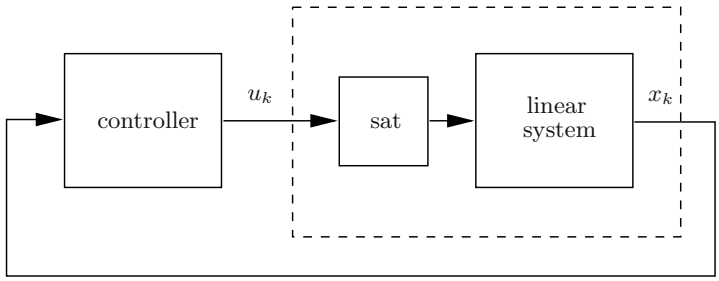

Figure 1.1. Feedback control loop for Example 1.2.1.

(i) Cautious Design

A *cautious* strategy would be, for example, to design a linear state feedback with *low gain* such that the control limits are never reached.

For example, using the objective function (1.1) with infinite horizon ($N = \infty$, $P = 0$) and weighting matrices $Q = C^{\mathrm{T}}C = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $R = 20$ gives the linear state feedback law:

$$u_k = -Kx_k = -\begin{bmatrix}0.1603 & 0.5662\end{bmatrix} x_k.$$

This control law *never violates the given physical limits* on the input *for the given initial condition*. The resulting input and output sequences are shown in Figure 1.2.

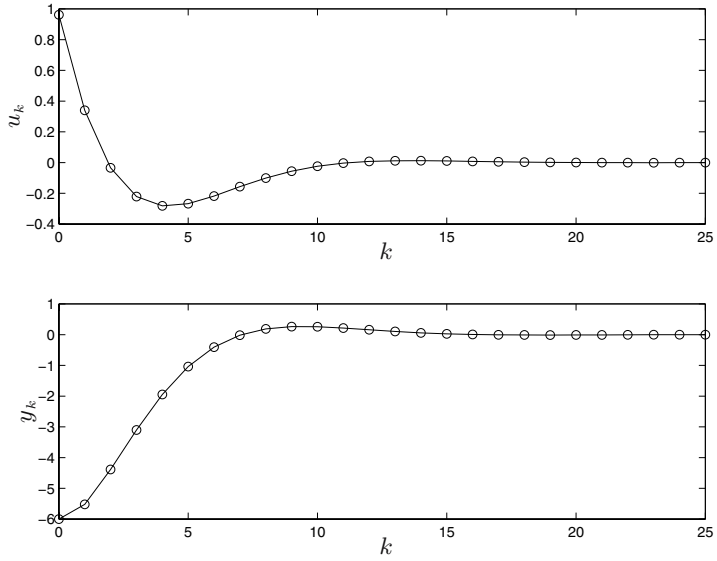

Figure 1.2. u_k and y_k for the cautious design $u_k = -Kx_k$ with weights $Q = C^{\mathrm{T}}C$ and $R = 20$.

We can see from Figure 1.2 that the input u_k has a maximum value close to 1 (achieved at $k = 0$) which clearly satisfies the given constraint *for this initial condition*. However, the achieved output response is rather slow. Indeed, it can be seen from Figure 1.2 that the "settling time" is of the order of eight samples.

(ii) Serendipitous Design

Now, suppose that for the same $Q = C^{\mathrm{T}}C$ in the infinite horizon objective function we try to obtain a faster response by reducing *the control weight to* $R = 2$. We expect that this will lead to a control law having "higher gain."

The resultant *higher gain control* would give the input and output sequences shown in dashed lines in Figure 1.3 *provided the input constraint could be removed* (that is, if the saturation block were removed from Figure 1.1). However, we can see that the *input constraints would have been violated* in the presence of actuator saturation. (The input at $k = 0$ is well beyond the allowed limit of $|u_k| = 1$.)

To *satisfy the constraints* we next incorporate the saturation function (1.3) in the controller and simply saturate the input signal when it violates the constraint. This leads to the control law:

$$u_k = \text{sat}(-Kx_k) = -\text{sat}(Kx_k).$$

Note that, in terms of performance, this is equivalent to simply letting the input saturate through the actuator in Figure 1.1. We call this control law *serendipitous* since no special considerations of the presence of the constraints have been made in the design calculations. The resulting input and output sequences are shown by circle-solid lines in Figure 1.3.

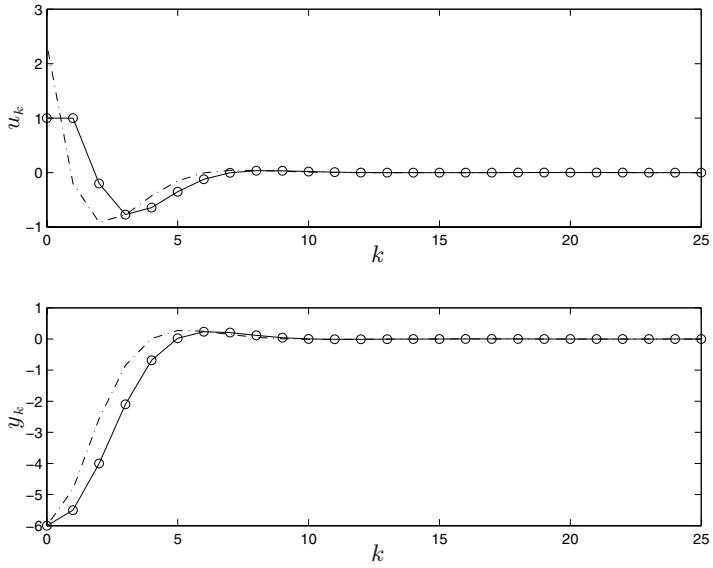

Figure 1.3. u_k and y_k for the unconstrained LQR design $u_k = -Kx_k$ (dashed line), and for the serendipitous strategy $u_k = -\text{sat}(Kx_k)$ (circle-solid line), with weights $Q = C^{\text{T}}C$ and $R = 2$.

We see from Figure 1.3 that the amount of overshoot is essentially the same whether or not the input is constrained. Of course, the response time achieved with a constrained input is longer than for the case when the input is unconstrained. However, note that the constraint is part of the physical reality and cannot be removed unless we replace the actuator. On the other hand, the serendipitous design (with $R = 2$) appears to be making better use of the available control authority than the cautious controller (with $R = 20$). Indeed, the settling time is now approximately five samples even when the input is constrained. This is approximately twice as fast as for the cautious controller, whose performance was shown in Figure 1.2.

Encouraged by the above result, we might be tempted to "push our luck" even further and aim for an even faster response by further reducing the weighting on the input signal. Accordingly, we *decrease the control weighting* in the LQR design even further, for example, to $R = 0.1$.

In Figure 1.4 we can see the resulting input and output sequences (when the input constraint, that is, the saturation block in Figure 1.1, is removed) for the linear controller $u_k = -Kx_k$ (dashed line). We now observe an *unconstrained* settling time of approximately three samples. However, when the input constraint is taken into account by setting $u_k = -\text{sat}(Kx_k)$, then we see that significant overshoot occurs and the settling time "blows out" to 12 samples (circle-solid line).

Perhaps we should not be surprised by this result since no special care has been taken to tailor the design to deal with constraints, that is, the approach remains serendipitous.

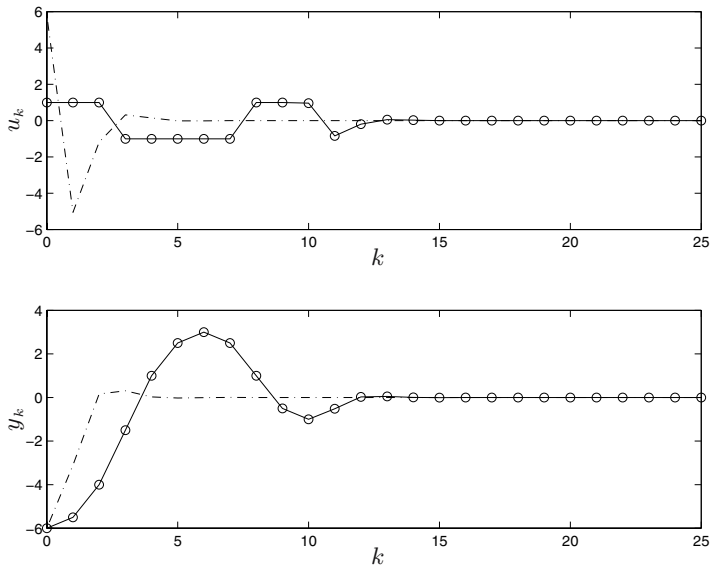

Figure 1.4. u_k and y_k for the unconstrained LQR design $u_k = -Kx_k$ (dashed line), and for the serendipitous strategy $u_k = -\text{sat}(Kx_k)$ (circle-solid line), with weights $Q = C^{\text{T}}C$ and $R = 0.1$.

We have seen above that as we try to push the system harder, the *serendipitous strategy ultimately fails to give a good result* leading to the output having large overshoot and long settling time. We can gain some insight into what has gone wrong by examining the state space trajectory corresponding to the serendipitous strategy. This is shown in Figure 1.5, where x_k^1 and x_k^2 denote the components of the state vector x_k in the discrete time model (1.2).

The control law $u = -\text{sat}(Kx)$ partitions the state space into three regions in accordance with the definition of the saturation function (1.3). Hence, the serendipitous strategy can be characterised as a *switched* control strategy in the following way:

$$u = \mathcal{K}(x) = \begin{cases} -Kx & \text{if } x \in R_0, \\ 1 & \text{if } x \in R_1, \\ -1 & \text{if } x \in R_2. \end{cases} \tag{1.4}$$

Notice that this is simply an alternative way of describing the serendipitous strategy since for $x \in R_0$ the input actually lies between the saturation limits. The partition is shown in Figure 1.5.

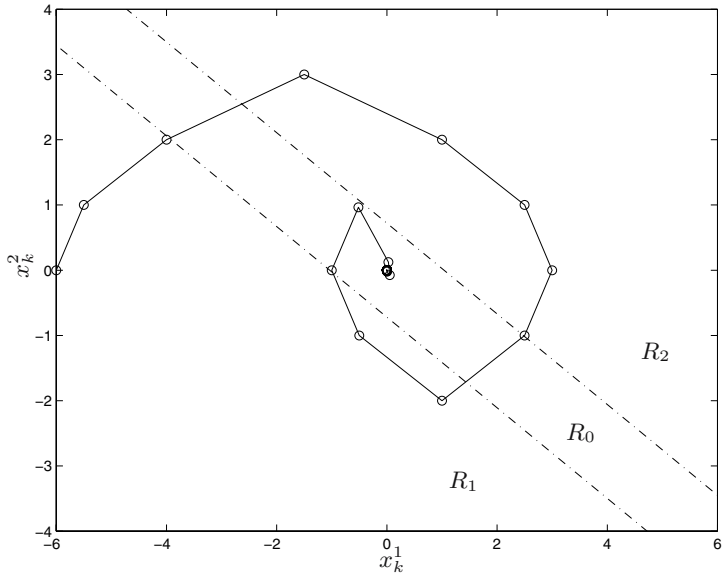

Figure 1.5. State space trajectory and space partition for the serendipitous strategy $u_k = -\text{sat}(Kx_k)$, with weights $Q = C^{\mathrm{T}}C$ and $R = 0.1$.

Examination of Figure 1.5 suggests a heuristic argument as to why the serendipitous control law may not be performing well in this case. We can think, in this example, of x^2 as "velocity" and x^1 as "position." Now, in our attempt to change the position rapidly (from -6 to 0), the velocity has been allowed to grow to a relatively high level ($+3$). This would be fine if the braking action were unconstrained. However, our input (including braking) is limited to the range $[-1,\ 1]$. Hence, the available braking is inadequate to "pull the system up", and overshoot occurs.

(iii) Tactical Design

Perhaps the above heuristic argument gives us some insight into how we could remedy the problem. A sensible idea would seem to be to try to "look ahead" and take account of future input constraints (that is, the limited braking authority available). To test this idea, we take the objective function (1.1) as a starting point.

We use a prediction horizon $N = 2$ and minimise, at each sampling instant i and for the current state x_i, the two-step objective function:

$$V_2(\{x_k\}, \{u_k\}) = \frac{1}{2} x_{i+2}^{\mathrm{T}} P x_{i+2} + \frac{1}{2} \sum_{k=i}^{i+1} \left(x_k^{\mathrm{T}} Q x_k + u_k^{\mathrm{T}} R u_k \right), \tag{1.5}$$

subject to the equality and inequality constraints:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, \\ |u_k| &\leq 1, \end{aligned} \tag{1.6}$$

for $k = i$ and $k = i + 1$.

In the objective function (1.5), we set, as before, $Q = C^{\mathrm{T}}C$, $R = 0.1$. The terminal state weighting matrix P is taken to be the solution of the Riccati equation $P = A^{\mathrm{T}}PA + Q - K^{\mathrm{T}}(R + B^{\mathrm{T}}PB)K$, where $K = (R + B^{\mathrm{T}}PB)^{-1}B^{\mathrm{T}}PA$ is the corresponding gain.

As a result of minimising (1.5) subject to (1.6), we obtain an optimal fixed-horizon control sequence $\{u_i, u_{i+1}\}$. We then apply the resulting value of u_i to the system. The state evolves to x_{i+1}. We now shift the time instant from i to $i + 1$ and repeat this procedure. This is called *receding horizon control* [RHC] or *model predictive control*. RHC has the ability to "look ahead" by considering the constraints not only at the current time i but also at future times within the prediction interval $[i, i + N - 1]$. (This idea will be developed in detail in Chapter 4.)

The input and output sequences for the LQR design $u = -Kx$ (dashed line) that violates the constraints and the sequences for the receding horizon design (circle-solid line) are shown in Figure 1.6.

We can see from Figure 1.6 that the output trajectory with constrained input now has minimal overshoot. Thus, the idea of "looking ahead" and applying the constraints in a receding horizon fashion has apparently "paid dividends."

Actually, we will see in Chapter 6 that the receding horizon strategy described above also leads to a partition of the state space into different regions in which affine control laws hold. The result is shown (for interest) in Figure 1.7. The region R_2 corresponds to the region R_2 in Figure 1.5 and represents the area of state space where $u = -1$ is applied. Comparing Figure 1.5 and Figure 1.7 we see that the region R_2 has been "bent over" in Figure 1.7 so that $u = -1$ occurs at lower values of x^2 (velocity) than was the case in

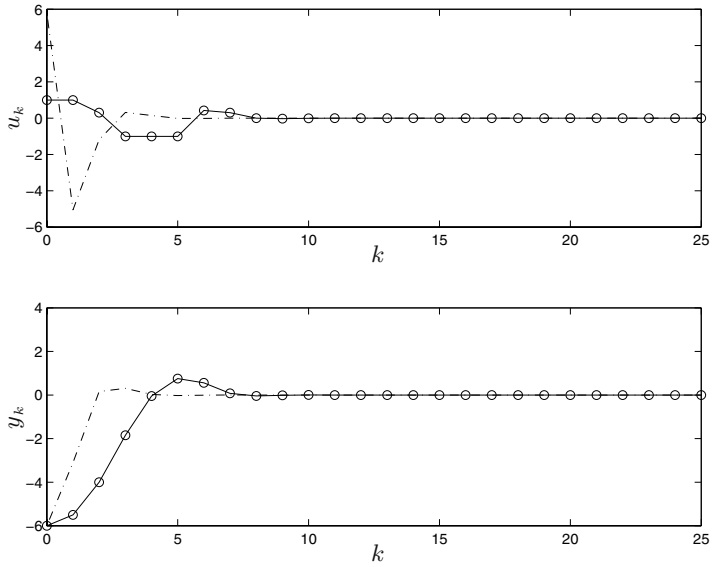

Figure 1.6. u_k and y_k for the unconstrained LQR design $u_k = -Kx_k$ (dashed line), and for the receding horizon design (circle-solid line), with weights $Q = C^{\mathrm{T}}C$ and $R = 0.1$.

Figure 1.5. This is in accordance with our heuristic argument about "needing to brake earlier." ○

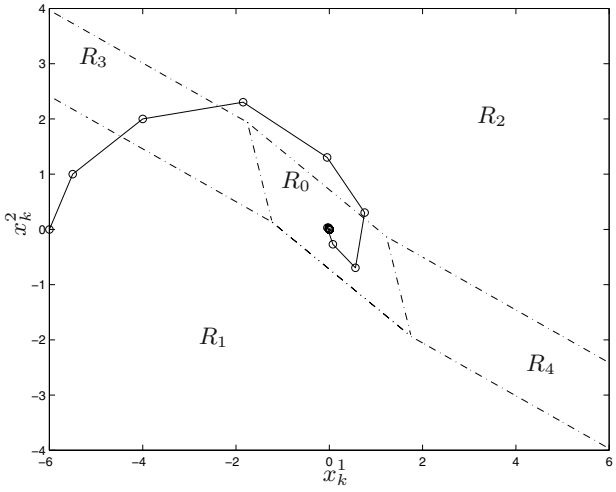

Figure 1.7. State space plot for the receding horizon tactical design.

Obviously we have not given full details of the above example especially in relation to the tactical approach; the example has been introduced only to "wet the readers' appetite" as to what might appear in the remainder of the book. Indeed, in forthcoming chapters, we will analyse, in some detail, the concepts raised in the above simple example.

1.3 Introduction to Constrained Estimation

Constraints are also often present in estimation problems. A classical example of a constrained estimation problem is the case in which *binary* data (say ± 1) are transmitted through a communication channel where it suffers dispersion causing the data to overlay itself. In the field of communications, this is commonly referred to as *intersymbol interference* [ISI]. The associated estimation problem is: Given the output of the channel, provide an estimate of the transmitted signal.

To illustrate some of the ideas involved in the above problem, let us assume, for simplicity, that the intersymbol interference produced by the channel can be modelled via a finite impulse response [FIR] model of the form:

$$y_k = \sum_{\ell=0}^{m} g_\ell u_{k-\ell} + n_k, \tag{1.7}$$

where $y_k,\ u_k,\ n_k$ denote the channel output, input and noise, respectively. Also, $(g_0 \dots g_m)$ denotes the (finite) impulse response of the channel. We assume here (for simplicity) that $g_0 \dots g_m$ are known. Also, for simplicity, we assume that the channel is minimum phase (that is, has a stable inverse).

Now, heuristically, one might expect that one should "invert" the channel so as to recover the input sequence $\{u_k\}$ from a given sequence of output data $\{y_k\}$. Such an inverse can be readily found by utilising feedback ideas. Specifically, if we expand the channel transfer function as:

$$G(z) = g_0 + \ldots + g_m z^{-m} = g_0 + \tilde{G}(z),$$

then we can form an inverse by the feedback circuit shown in Figure 1.8.

To verify that the circuit of Figure 1.8 does, indeed, produce an inverse, we see that the transfer function from y_k to $\tilde{u}_k$ is

$$T(z) = \frac{\dfrac{1}{g_0}}{1 + \dfrac{\tilde{G}(z)}{g_0}} = \frac{1}{g_0 + \tilde{G}(z)} = \frac{1}{G(z)}.$$

Thus, we have generated an inverse to the system transfer function $G(z)$. Hence, in the absence of noise and other errors, we can expect that the signal $\tilde{u}_k$ in Figure 1.8 will converge to u_k following an initial transient (note that

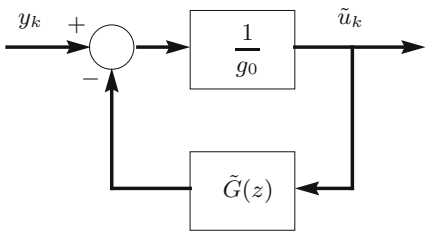

Figure 1.8. Feedback inverse circuit.

we have assumed that $G(z)$ has a stable inverse). Under ideal conditions this is exactly what does happen. However, in practice, the presence of the noise term in (1.7) will lead to estimation errors. Indeed, a little thought shows that $\tilde{u}_k$ may not even belong to the set $\{+1, -1\}$ even though we know, a priori, that the true transmitted signal, u_k, does.

An improvement seems to be to simply take the nearest value from the set $\{+1,\ -1\}$ corresponding to $\tilde{u}_k$. This leads to the circuit shown in Figure 1.9, where

$$\text{sign}(\tilde{u}_k) \triangleq \begin{cases} +1 & \text{if } \tilde{u}_k \geq 0, \\ -1 & \text{if } \tilde{u}_k < 0. \end{cases}$$

Comparing Figure 1.9 with Figure 1.8 may lead us to develop a further

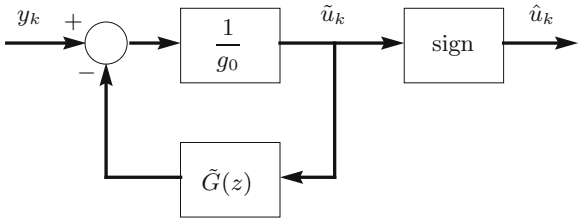

Figure 1.9. Constrained feedback inverse circuit.

embellishment of this simple idea. In particular, we see in Figure 1.8 that the feedback path through the transfer function $\tilde{G}(z)$ uses the estimated input $\tilde{u}_k$. Now, in Figure 1.9 our belief is that $\hat{u}_k$ should be a better estimate of the input than $\tilde{u}_k$ since we have *forced* the constraint $\hat{u}_k \in \{+1, -1\}$. This suggests that we could try feeding back $\hat{u}_k$ instead of $\tilde{u}_k$, as shown in Figure 1.10.

The arguments leading to Figure 1.10 are rather heuristic. Nonetheless, the constrained estimator in Figure 1.10 finds very widespread application in

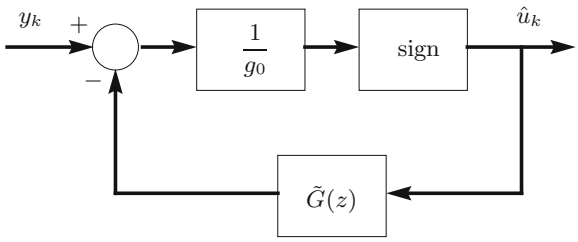

Figure 1.10. Constrained estimation with decision feedback, or "decision feedback equaliser [DFE]."

digital communications, where it is given a special name—decision feedback equaliser [DFE].

The reader might now be asking how one could improve on the circuit of Figure 1.10. We can gain some insight as to from where further improvements might come by expressing the result shown in Figure 1.10 as the solution to an optimisation problem. Specifically, assume that we are given (estimates of) past values of the input, $\{\hat{u}_{k-1}, \ldots, \hat{u}_{k-m}, \ldots\}$, and that we model the output $\hat{y}_k$ as

$$\hat{y}_k = g_0 u'_k + g_1 \hat{u}_{k-1} + \ldots + g_m \hat{u}_{k-m}.$$

We can now ask what value of u'_k causes $\hat{y}_k$ to be, at time k, as close as possible to the observed output y_k. We measure how close $\hat{y}_k$ is to y_k by the following one-step objective function:

$$V_1(\hat{y}_k, u'_k) = [y_k - \hat{y}_k]^2.$$

We also require that $u'_k \in \{+1, -1\}$. The solution to this constrained optimisation problem is readily seen to be:

$$\hat{u}_k = \text{sign}\left\{\frac{1}{g_0}[y_k - g_1 \hat{u}_{k-1} - \ldots - g_m \hat{u}_{k-m}]\right\}. \tag{1.8}$$

However, the reader can verify that this precisely corresponds to the arrangement illustrated in Figure 1.10. One might anticipate that by exploiting the connection with constrained optimisation one can obtain better performance, since more elaborate objective functions can be employed. How this might be achieved is discussed below, and a more detailed description will be given in Chapter 13.

The following example illustrates the above ideas.

Example 1.3.1. Consider the channel model

$$y_k = u_k - 1.7u_{k-1} + 0.72u_{k-2} + n_k,$$

where u_k is a random binary signal and n_k is an independent identically distributed [i.i.d.] noise having a Gaussian distribution of variance σ^2. We first assume the ideal situation in which the channel has no noise, $\sigma^2 = 0$. Since the channel model has a stable inverse, we implement the inversion estimator depicted in Figure 1.8. The result of the simulation is represented in Figure 1.11. Notice that the estimator yields perfect signal recovery. This

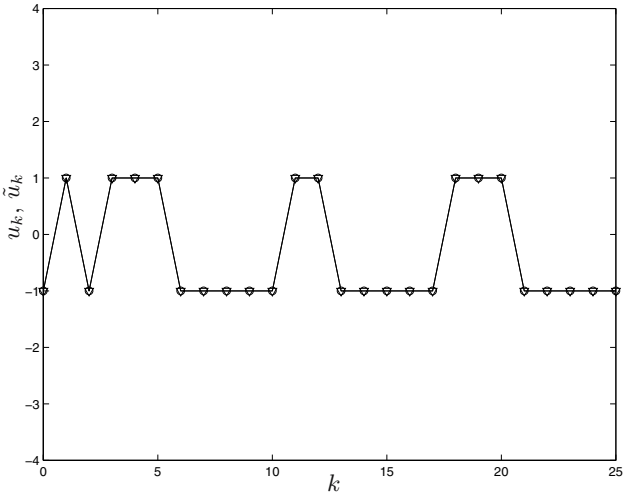

Figure 1.11. Data u_k (circle-solid line) and estimate $\tilde{u}_k$ (triangle-solid line) using the feedback inverse circuit of Figure 1.8. Noise variance: $\sigma^2 = 0$.

is very encouraging. However, this situation in which no noise is present is unrealistic.

Next, we simulate the inversion estimator of Figure 1.8 when the received signal is affected by noise n_k of variance $\sigma^2 = 0.1$. The result of the simulation is shown in Figure 1.12. Note that in this case the estimate $\tilde{u}_k$ differs from u_k and does not belong to the range ± 1. We conclude that not taking account of the constraints in the estimation leads to a poor result.

We next simulate the estimator represented in Figure 1.9 with noise of variance $\sigma^2 = 0.1$. In this implementation, the nearest value of the estimate of the previous scheme from the set $\{+1, -1\}$ is taken. The result is shown in Figure 1.13. It can be observed that now the estimate $\hat{u}_k$ belongs to the set $\{+1, -1\}$, but the result is still poor. The reason is that we have not "informed" the estimator about the constrained estimates but have simply applied the constraint $\hat{u}_k = \text{sign}(\tilde{u}_k) \in \{+1, -1\}$ "after the event."

As a further improvement to our estimator, we next implement the estimator of Figure 1.10 (the DFE) where we now feed back $\hat{u}_k \in \{+1, -1\}$ instead of $\tilde{u}_k$, thereby informing the estimator about the presence of constraints. The

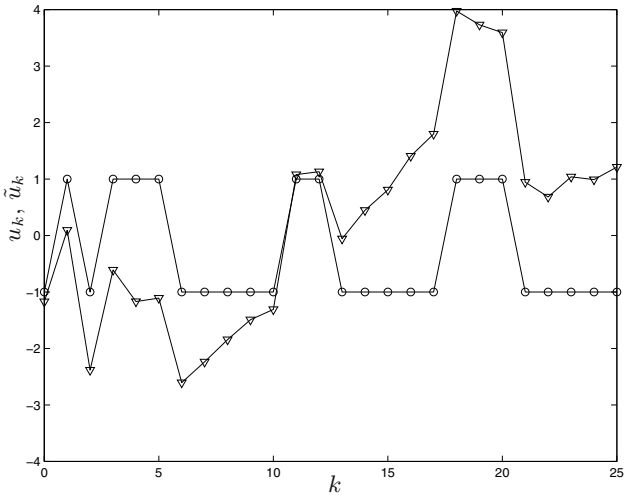

Figure 1.12. Data u_k (circle-solid line) and estimate $\tilde{u}_k$ (triangle-solid line) using the feedback inverse circuit of Figure 1.8. Noise variance: $\sigma^2 = 0.1$.

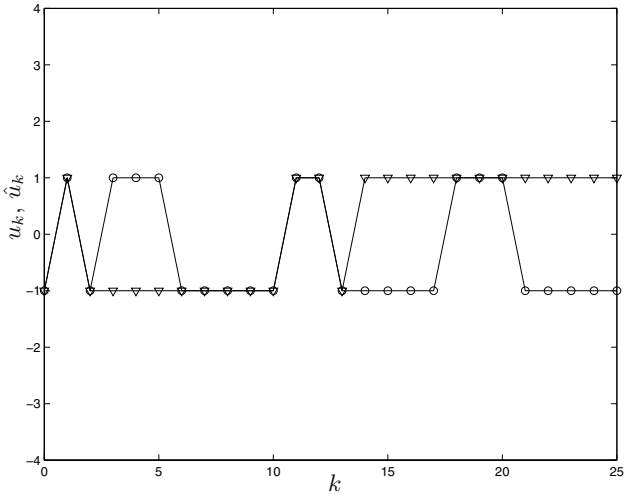

Figure 1.13. Data u_k (circle-solid line) and estimate $\hat{u}_k$ (triangle-solid line) using the constrained feedback inverse circuit of Figure 1.9. Noise variance: $\sigma^2 = 0.1$.

result of the simulation, for a noise of variance $\sigma^2 = 0.1$, is shown in Figure 1.14. Note that, despite the presence of noise in the channel, the DFE recovers the signal perfectly.

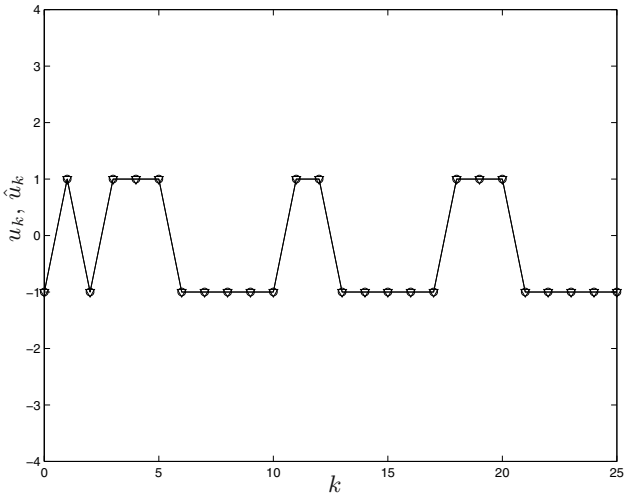

Figure 1.14. Data u_k (circle-solid line) and estimate $\hat{u}_k$ (triangle-solid line) using the DFE of Figure 1.10. Noise variance: $\sigma^2 = 0.1$.

One might wonder if the DFE circuit would always perform so well. We next investigate the performance of the DFE of Figure 1.10 when the noise variance is increased by a factor of 2; that is, $\sigma^2 = 0.2$. The result of the simulation is shown in Figure 1.15. Note that we have poor performance. The

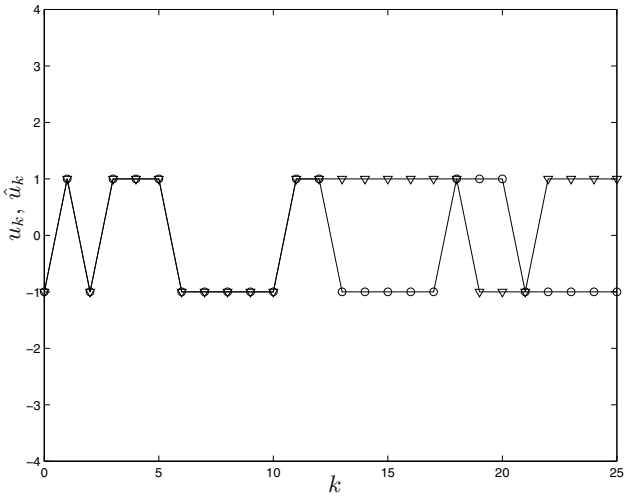

Figure 1.15. Data u_k (circle-solid line) and estimate $\hat{u}_k$ (triangle-solid line) using the DFE of Figure 1.10. Noise variance: $\sigma^2 = 0.2$.

reason is that we are only considering one observation at a time (as mentioned earlier, this scheme is equivalent to solving a one-step optimisation problem). It is a well-known phenomenon with this circuit that, once a detection error occurs, it may propagate. Thus, errors typically occur in "bursts." The reason for this error propagation is that previous estimates are assumed to be equal to the true signal. As we will study later in Chapter 13, other estimation mechanisms can be implemented in which multiple observations are considered simultaneously and some "degree of belief" in previous estimates is incorporated. We will give a first taste of these ideas below. ∘

The reader has probably noticed that there is a very close connection between the above problem and the tactical approach to the control problem discussed in Section 1.2. This suggests a route by which we may be able to improve the estimate given in (1.8). Referring to Example 1.2.1, we found that looking ahead so as to account for future consequences of current actions was helpful. Thus, we might be led to ask what would happen if we did not fix u'_k based only on the observation y_k but waited until we had observed both y_k and y_{k+1}. Of course, y_{k+1} also depends on u_{k+1}, but this consideration could be dealt with by asking that values of u'_k and u'_{k+1} belonging to the set $\{+1, -1\}$ be chosen such that the following two-stage objective function is minimised:

$$V_2(\hat{y}_k, \hat{y}_{k+1}, u'_k, u'_{k+1}) = [y_k - \hat{y}_k]^2 + [y_{k+1} - \hat{y}_{k+1}]^2, \tag{1.9}$$

where

$$\hat{y}_k = g_0 u'_k + g_1 \hat{u}_{k-1} + \ldots + g_m \hat{u}_{k-m}, \tag{1.10}$$

$$\hat{y}_{k+1} = g_0 u'_{k+1} + g_1 u'_k + g_2 \hat{u}_{k-1} + \ldots + g_m \hat{u}_{k-m+1}, \tag{1.11}$$

and where the past estimates $\{\hat{u}_{k-1}, \hat{u}_{k-2}, \ldots\}$ are again assumed fixed and known.

The solution to the above problem can be readily computed by simple evaluation of V_2 for all possible constrained inputs; that is, for

$$\{u'_k, u'_{k+1}\} \in \big\{\{-1,-1\}, \{-1,1\}, \{1,1\}, \{1,-1\}\big\}. \tag{1.12}$$

Notice that there are four possibilities and the optimal solution is simply the one that yields the lowest value of V_2. We could then fix the estimate of u_k (denoted $\hat{u}_k$) as the first element of the solution to this optimisation problem. We might then proceed to measure y_{k+2} and re-estimate u_{k+1}, plus obtain a fresh estimate of u_{k+2} by minimising:

$$V_2(\hat{y}_{k+1}, \hat{y}_{k+2}, u'_{k+1}, u'_{k+2}) = [y_{k+1} - \hat{y}_{k+1}]^2 + [y_{k+2} - \hat{y}_{k+2}]^2,$$

where

$$\hat{y}_{k+1} = g_0 u'_{k+1} + g_1 \hat{u}_k + \ldots + g_m \hat{u}_{k-m+1},$$
$$\hat{y}_{k+2} = g_0 u'_{k+2} + g_1 u'_{k+1} + g_2 \hat{u}_k + \ldots + g_m \hat{u}_{k-m+2},$$

and where $\{\hat{u}_k, \hat{u}_{k-1}, \ldots\}$ are now assumed fixed and known.

By the above procedure, we are already generating constrained estimates via a *moving horizon estimator* [MHE] subject to the constraint $u'_k \in \{+1, -1\}$. This kind of estimator will be studied in detail in Chapter 13.

Example 1.3.2. Consider again the channel model of Example 1.3.1. Here we implement a moving horizon estimator as described above. That is, we minimise, at each step, the two-stage objective function (1.9), subject to (1.10)–(1.11) and the constraints (1.12). We then take as the current estimate $\hat{u}_k$ the first value u'_k of the minimising sequence $\{u'_k, u'_{k+1}\}$.

The corresponding simulation results, for noise variance $\sigma^2 = 0.2$, are shown in Figure 1.16. We can see from this figure that the estimator recovers the signal perfectly. Comparing with Figure 1.15 (which shows the estimate provided by the DFE for the same noise variance), we can see that "looking ahead" two steps has been beneficial in this case. ○

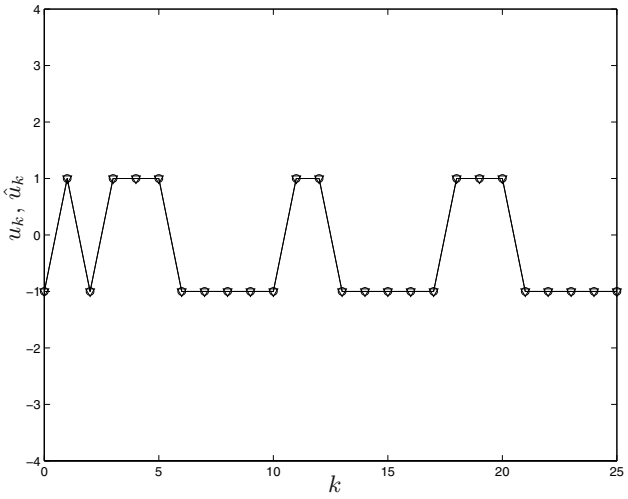

Figure 1.16. Data u_k (circle-solid line) and estimate $\hat{u}_k$ (triangle-solid line) using the moving horizon two-step estimator. Noise variance: $\sigma^2 = 0.2$.

1.4 Connections Between Constrained Control and Estimation

The brief introduction to constrained control and estimation given in Section 1.2 and Section 1.3 will have, no doubt, left the reader with the impression that these two problems are, at least, very *similar*. Indeed, both have been

cast as finite horizon constrained optimisation problems. We will see later that these problems lead to the same underlying question, the only difference being a rather minor issue associated with the boundary conditions. Actually, we will show that a strong connection between constrained control and estimation problems is revealed when looked upon via a *Lagrangian duality* perspective. This will be the topic of Chapter 10.

1.5 The Remainder of the Book

The remainder of the book is devoted to expanding on the ideas introduced above. We will emphasise constrained optimisation approaches to the topics of control and estimation. Thus, we begin in the next chapter with a review of basic optimisation theory. This will be followed in Chapter 3 by a review of classical optimal control theory, including the discrete minimum principle. In Chapter 4, and following chapters, we will apply these ideas to the specific issues that arise in control and estimation problems.

1.6 Further Reading

For complete list of references cited, see References section at the end of book.

General

An introduction to unconstrained (linear) control can be found in a host of textbooks, such as Anderson and Moore (1989), Åström and Wittenmark (1990), Bitmead, Gevers and Wertz (1990), Goodwin, Graebe and Salgado (2001), Zhou, Doyle and Glover (1996).

The following books complement the material presented in the current chapter and give further information on receding horizon control: Camacho and Bordons (1999), Maciejowski (2002), Borrelli (2003), Rossiter (2003).

The book Proakis (1995) gives further background on channel equalisation in digital communications. See also the survey papers Qureshi (1985), Tugnait, Tong and Ding (2000).