2

Discrete-time MPC with Constraints

2.1 Introduction

This chapter discusses discrete-time model predictive control with constraints. The chapter begins with a motivational example to illustrate how the performance of a control system can deteriorate significantly when the control signals from the original design meet with operational constraints. The example also shows that with a small modification, the degree of performance deterioration can be reduced if the constraints are incorporated in the implementation, leading to the idea of constrained control. Then, the chapter reveals how to formulate the constrained control problem in the context of predictive control, which essentially becomes a quadratic programming problem. Assuming that most readers have not studied quadratic programming before, a section is devoted to introducing the fundamentals of quadratic programming and presenting the solutions with simple and effective numerical algorithms. The final section of this chapter shows several examples of constrained control problems.

2.2 Motivational Examples

Before we begin our study on constrained control, let us look at an example where the control system operates with and without control signal saturation limits.

Example 2.1. A mathematical model for an undamped oscillator is given by

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 0 \; 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \quad (2.1)$$

With sampling interval $\Delta t = 0.1$, the corresponding discrete-time state-space model is

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9801 & 0.0993 \\ -0.3973 & 0.9801 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0993 \\ -0.0199 \end{bmatrix} u(k) \quad (2.2)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \quad (2.3)$$

Suppose that the design objective is to design the predictive control system such that the output of the plant has to track a unit step reference signal as fast as possible. To this end, we select the prediction horizon $N_p = 10$ and the control horizon $N_c = 3$. There is no weight on the control signal, *i.e.*, $\bar{R} = 0$. Examine what happens if the control amplitude is limited to ± 25 by saturation.

Solution. The data matrices in the predictive control system are

$$\Phi^T \Phi = \begin{bmatrix} 6.0067 & 4.8853 & 3.8150 \\ 4.8853 & 4.0013 & 3.1475 \\ 3.8150 & 3.1475 & 2.4952 \end{bmatrix}; \Phi^T F = \begin{bmatrix} 65.5285 & -25.2099 & -6.1768 \\ 53.1281 & -19.6709 & -4.7606 \\ 41.3553 & -14.6974 & -3.5334 \end{bmatrix}$$

$$\Phi^T \bar{R}s = \begin{bmatrix} -6.1768 \\ -4.7606 \\ -3.5334 \end{bmatrix}.$$

The state feedback gain matrix is

$$K_{mpc} = \begin{bmatrix} 17.9064 & -39.0664 & -29.9659 \end{bmatrix}.$$

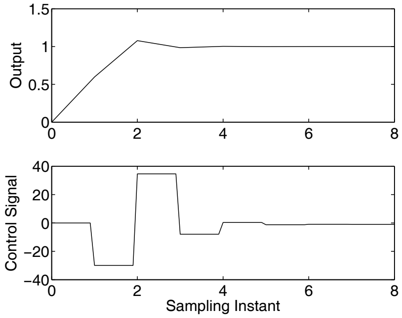
The closed-loop eigenvalues are at $-0.1946, 0, 0$. An observer is designed with poles positioned at $0, 0, 0$. In the following, we look at two cases: without control saturation; and with control signal saturation.

Case A. Without control saturation

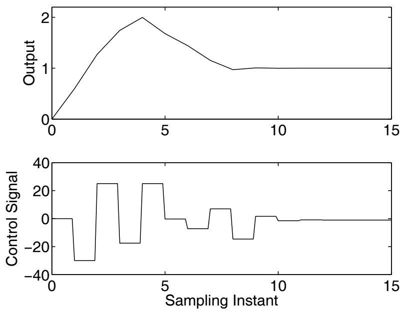
The closed-loop response is illustrated in Figure 2.1a. It is seen that the output converges to the set-point signal after 4 samples. Indeed, the design objective has been achieved. If the control amplitude is of concern, then we note that this optimal control has a large amplitude that is close to 40 at its maximum.

Case B. With control saturation

Assume that the control amplitude has limits at ± 25 due to operational constraint, namely, $-25 \leq u(k) \leq 25$. Then, this limit prevents the control signal from being implemented to the plant when its amplitude exceeds this limit. Thus, $u(k) = 25$, if $u(k) > 25$; and $u(k) = -25$ if $u(k) < -25$. When this



(a) Closed-loop response without constraint



(b) Closed-loop control with saturation

Fig. 2.1. Comparison of responses with and without constraints

happens, the closed-loop performance significantly deteriorates, as shown in Figure 2.1b. The figure shows that when the control signal comes out of the saturation after two sample periods, it becomes oscillatory, and as a result, the plant output has a significant over-shoot.

This example illustrated that if we do not pay attention to the saturation of the control, then in the presence of constraints, the closed-loop control performance could severely deteriorate. From this example, it is obvious that it is important to find a way to deal with the problem when the control signal becomes saturated. The next example shows that a small modification in the predictive control law will enable the system to handle the constraint without significant performance deterioration.

Example 2.2. A common practice in dealing with saturation is to let the model know the difference in $\Delta u(k)$ when saturation becomes effective. Continue with Example 2.1 with a modification of the control calculation where the difference of the control signal $\Delta u(k)$ is taken into consideration in the presence of the constraint.

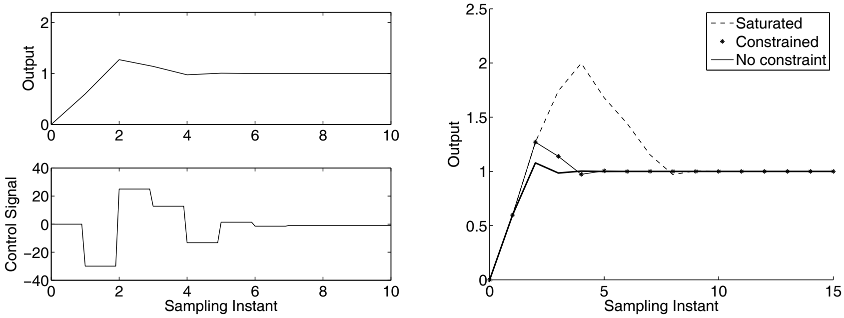
Solution. The small modification is to calculate $\Delta u(k_i)$ in the following way. If the calculated control $u(k_i) > 25$, then

$$\begin{aligned} u(k_i) &= 25 \;\; and \\ \Delta u(k_i) &= 25 - u(k_i - 1). \end{aligned} \tag{2.4}$$

If the calculated control $u(k_i) < -25$, then

$$\begin{aligned} u(k_i) &= -25 \;\; and \\ \Delta u(k_i) &= -25 - u(k_i - 1). \end{aligned} \tag{2.5}$$

This new $\Delta u(k_i)$ is used in the observer to predict the next sample of state variable $\hat{x}(k_i + 1)$:



(a) Saturation with modification (b) Output responses for three cases

Fig. 2.2. Comparison of results when constraints are present

$$\hat{x}(k_i + 1) = A\hat{x}(k_i) + B\Delta u(k_i) + K_{ob}(y(k_i) - C\hat{x}(k_i)). \tag{2.6}$$

The calculation is performed for all sample points. This small modification results in a significant change in the control system performance. Figure 2.2a shows the improvement of the control system in the presence of activated constraint. What we notice is that the control signal comes out of saturation without oscillation. As a result, the over-shoot in the closed-loop response (as shown in Figure 2.2b of Example 2.1) is significantly reduced. For comparison purpose, the output responses are presented in Figure 2.2b for three different cases, where Case A is the output response without saturation (solid line); Case B is the output response with saturation (dotted); and Case C is the output response with modified control saturation (dotted+solid). The improvement on closed-loop performance with this modified saturation is further illustrated by the comparative data in Table 2.1. As we will learn later in the chapter, this solution is in fact the optimal solution in the context of predictive control with amplitude constraint for a single-input, single-output system.

There are several issues to note from these examples. The saturation on the control signal can make the control system performance deteriorate signifi-

Table 2.1. Comparison of performance parameters. Predictive control system with saturation and predictive control system with modified saturation

	With control saturation	With modified control saturation
y_{max}	2.01	1.25
Over-shoot(%)	100	25
No. Samples to settling	11	6
u^{max}	25	25
u^{min}	-25	-25

cantly. The modification to overcome control saturation effects is to calculate the value $\Delta u(k)$ when the saturation is reached, and use this information to modify the predicted state variables. Because it was a single-input, single-output system and only two constraints were imposed, this small modification was feasible. However, for a multi-input, multi-output system, the limits of the system operation appear in many forms, such as the limits on each control signal and its difference $\Delta u(k)$, as well as on the state variables and output variables. It is a much more complex task to work out the individual saturation limits in a co-ordinated manner. Even if we could do this, the solution still would not guarantee the optimal performance of the multi-input, multi-output system. It is on these grounds that we propose a constrained control framework using the model predictive control system. The strength of the approach lies in the optimality that it achieves in a systematic manner, and the flexibility/generality to cope with a multi-input, multi-out system with various constraints. Perhaps above all, this simplicity in concept is readily accepted by application engineers.

2.3 Formulation of Constrained Control Problems

The core idea in Example 2.2 was to modify $\Delta u(k)$ to suit the situation when the constraint became activated. In the context of predictive control, this problem is handled systematically by using optimization. To this end, we need to formulate the predictive control problem as an optimization problem that takes into account the constraints present.

This section discusses the operational constraints that are frequently encountered in the design of control systems. These operational constraints are presented as linear inequalities of the control and plant variables.

2.3.1 Frequently Used Operational Constraints

There are three major types of constraints frequently encountered in applications. The first two types deal with constraints imposed on the control variables $u(k)$, and the third type of constraint deals with output $y(k)$ or state variable $x(k)$ constraints. For clarity, we will discuss single-input, single-output systems first and subsequently extend the cases to multi-input, multi-output systems.

Constraints on the Control Variable Incremental Variation

These are hard constraints on the size of the control signal movements, *i.e.*, on the rate of change of the control variables ($\Delta u(k)$). Suppose that for a single-input system, the upper limit is Δu^{max} and the lower limit is Δu^{min}. The constraints are specified in the form

$$\Delta u^{min} \leq \Delta u(k) \leq \Delta u^{max}. \tag{2.7}$$

Note that we use *less than plus equal to* in (2.7), where the equality will play the critical role in the solution of the constrained control problem (see later sections).

The rate of change constraints can be used to impose directional movement constraints on the control variables; for instance, if $u(k)$ can only increase, not decrease, then possibly, we select $0 \leq \Delta u(k) \leq \Delta u^{max}$. The constraint on $\Delta u(k)$ can be used to cope with the cases where the rate of change of the control amplitude is restricted or limited in value. For example, in a control system implementation, assuming that the control variable $u(k)$ is only permitted to increase or decrease in a magnitude less 0.1 unit, then the operational constraint is

$$-0.1 \leq \Delta u(k) \leq 0.1.$$

Constraints on the Amplitude of the Control Variable

These are the most commonly encountered constraints among all constraint types. For instance, we cannot expect a valve to open more than 100 percent nor a voltage to go beyond a given range. These are the physical hard constraints on the system. Simply, we demand that

$$u^{min} \leq u(k) \leq u^{max}.$$

Here, we need to pay particular attention to the fact that $u(k)$ is an incremental variable, not the actual physical variable. The actual physical control variable equals the incremental variable u plus its steady-state value u_{ss}. A common mistake is to mix these two. For instance, if a valve is allowed to open in the range between 15% and 80% and the valve's normal operating value is at 30%, then $u^{min} = 15\% - 30\% = -15\%$ and $u^{max} = 80\% - 30\% = 50\%$.

Output Constraints

We can also specify the operating range for the plant output. For instance, supposing that the output $y(k)$ has an upper limit y^{max} and a lower limit y^{min}, then the output constraints are specified as

$$y^{min} \leq y(k) \leq y^{max}. \tag{2.8}$$

Output constraints are often implemented as 'soft' constraints in the way that a slack variable $s_v > 0$ is added to the constraints, forming

$$y^{min} - s_v \leq y(k) \leq y^{max} + s_v. \tag{2.9}$$

There is a primary reason why we use a slack variable to form 'soft' constraints for output. Output constraints often cause large changes in both the control and incremental control variables when they are enforced (we term them *become active* in the later sections). When that happens, the control or incremental control variables can violate their own constraints and the problem of constraint conflict occurs. In the situations where the constraints on the control variables are more essential to plant operation, the output constraints are often relaxed by selecting a larger slack variable s_v to resolve the conflict problem.

Similarly, we can impose constraints on the state variables if they are measurable or impose the constraints on observer state variables. They also need to be in the form of 'soft' constraints for the same reasons as the output case above.

Constraints in a Multi-input and Multi-output Setting

If there is more than one input, then the constraints are specified for each input independently. In the multi-input case, suppose that the constraints are given for the upper limits as

$$\begin{bmatrix} \Delta u_1^{max} & \Delta u_2^{max} & \ldots & \Delta u_m^{max} \end{bmatrix},$$

and lower limits as

$$\begin{bmatrix} \Delta u_1^{min} & \Delta u_2^{min} & \ldots & \Delta u_m^{min} \end{bmatrix}.$$

Each variable with rate of change is specified as

$$\begin{aligned} \Delta u_1^{min} &\leq \Delta u_1(k) \leq \Delta u_1^{max} \\ \Delta u_2^{min} &\leq \Delta u_2(k) \leq \Delta u_2^{max} \\ &\vdots \\ \Delta u_m^{min} &\leq \Delta u_m(k) \leq \Delta u_m^{max}. \end{aligned} \tag{2.10}$$

Similarly, suppose that the constraints are given for the upper limit of the control signal as

$$\begin{bmatrix} u_1^{max} & u_2^{max} & \ldots & u_m^{max} \end{bmatrix},$$

and lower limit as

$$\begin{bmatrix} u_1^{min} & u_2^{min} & \ldots & u_m^{min} \end{bmatrix}.$$

Then, the amplitude of each control signal is required to satisfy the constraints:

$$\begin{aligned} u_1^{min} &\leq u_1(k) \leq u_1^{max} \\ u_2^{min} &\leq u_2(k) \leq u_2^{max} \\ &\vdots \\ u_m^{min} &\leq u_m(k) \leq u_m^{max}. \end{aligned} \tag{2.11}$$

Similarly, constraints are specified for each output and state variable if they are required. In short, the constraints for a multi-input and multi-output system are specified for each input and output independently.

2.3.2 Constraints as Part of the Optimal Solution

Having formulated the constraints as part of the design requirements, the next step is to translate them into linear inequalities, and relate them to the original model predictive control problem. The key here is to parameterize the constrained variables using the same parameter vector ΔU as the ones used in the design of predictive control. Therefore, the constraints are expressed in a set of linear equations based on the parameter vector ΔU. The vector ΔU is often called the *decision variable* in optimization literature. Since the predictive control problem is formulated and solved in the framework of receding horizon control, the constraints are taken into consideration for each moving horizon window. This allows us to vary the constraints at the beginning of each optimization window, and also gives us the means to tackle the constrained control problem numerically. Based on this idea, if we want to impose the constraints on the rate of change of the control signal $\Delta u(k)$ at time k_i, the constraints at sample time k_i are expressed as

$$\Delta u^{min} \leq \Delta u(k_i) \leq \Delta u^{max}.$$

From the time instance k_i, the predictive control scheme looks into the future. The constraints at future samples, for example on the first three samples, $\Delta u(k_i)$,$\Delta u(k_i+1)$, $\Delta u(k_i+2)$ are imposed as

$$\Delta u^{min} \leq \Delta u(k_i) \leq \Delta u^{max}$$
$$\Delta u^{min} \leq \Delta u(k_i+1) \leq \Delta u^{max}$$
$$\Delta u^{min} \leq \Delta u(k_i+2) \leq \Delta u^{max}.$$

In principle, all the constraints are defined within the prediction horizon. However, in order to reduce the computational load, we sometimes choose a smaller set of sampling instants at which to impose the constraints, instead of all the future samples. The following example shows how to express the constraints from the design specification in terms of a function of ΔU.

Example 2.3. In the motor control system, suppose that the input voltage variation is limited to 2 V and 6 V. The steady state of the control signal is at 4 V. Assuming that the control horizon is selected to be $N_c = 4$, express the constraint on $\Delta u(k_i)$ and $\Delta u(k_i+1)$ in terms of ΔU for the first two sample times.

Solution. The parameter vector to be optimized in the predictive control system at time k_i is $\Delta U = \left[\Delta u(k_i)\ \Delta u(k_i+1)\ \Delta u(k_i+2)\ \Delta u(k_i+3)\right]^T$.

Note that

$$u(k_i) = u(k_i-1) + \Delta u(k_i) = u(k_i-1) + \left[1\ 0\ 0\ 0\right]\Delta U \tag{2.12}$$

$$\begin{aligned} u(k_i+1) &= u(k_i) + \Delta u(k_i+1) = u(k_i-1) + \Delta u(k_i) + \Delta u(k_i+1) \\ &= u(k_i-1) + \left[1\ 1\ 0\ 0\right]\Delta U. \end{aligned} \tag{2.13}$$

With the limits on the control variables, by subtracting the steady-state value of the control, as $u^{min} = 2 - 4 = -2$ and $u^{max} = 6 - 4 = 2$, the constraints are expressed as

$$\begin{bmatrix} -2 \\ -2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(k_i - 1) + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \Delta U \leq \begin{bmatrix} 2 \\ 2 \end{bmatrix}. \tag{2.14}$$

Now, since we have expressed the constraints as the inequalities with linear-in-the-parameter ΔU, the next step is to combine the constraints with the original cost function J used in the design of predictive control. As the optimal solutions will be obtained using quadratic programming, the constraints need to be decomposed into two parts to reflect the lower limit, and the upper limit with opposite sign. Namely, for instance, the constraints

$$\Delta U^{min} \leq \Delta U \leq \Delta U^{max}$$

will be expressed by two inequalities:

$$-\Delta U \leq -\Delta U^{min} \tag{2.15}$$
$$\Delta U \leq \Delta U^{max}. \tag{2.16}$$

In a matrix form, this becomes

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}. \tag{2.17}$$

This procedure applies to all the constraints mentioned in this section, including control and output constraints.

Traditionally, the constraints are imposed for all future sampling instants, and all constraints are expressed in terms of the parameter vector ΔU. In the case of a manipulated variable constraint, we write:

$$\begin{bmatrix} u(k_i) \\ u(k_i+1) \\ u(k_i+2) \\ \vdots \\ u(k_i+N_c-1) \end{bmatrix} = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix} u(k_i-1) + \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ I & I & 0 & \dots & 0 \\ I & I & I & \dots & 0 \\ \vdots \\ I & I & \dots & I & I \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \\ \vdots \\ \Delta u(k_i+N_c-1) \end{bmatrix}. \tag{2.18}$$

Re-writing (2.18) in a compact matrix form, with C_1 and C_2 corresponding to the appropriate matrices, then the constraints for the control movement are imposed as,

$$-(C_1 u(k_i-1) + C_2 \Delta U) \leq -U^{min} \tag{2.19}$$
$$(C_1 u(k_i-1) + C_2 \Delta U) \leq U^{max}, \tag{2.20}$$

where U^{min} and U^{max} are column vectors with N_c elements of u^{min} and u^{max}, respectively. Similarly, for the increment of the control signal, we have the constraints:

$$-\Delta U \leq -\Delta U^{min} \tag{2.21}$$
$$\Delta U \leq \Delta U^{max}, \tag{2.22}$$

where ΔU^{min} and ΔU^{max} are column vectors with N_c elements of Δu^{min} and Δu^{max}, respectively. The output constraints are expressed in terms of ΔU:

$$Y^{min} \leq Fx(k_i) + \Phi \Delta U \leq Y^{max}. \tag{2.23}$$

Finally, the model predictive control in the presence of hard constraints is proposed as finding the parameter vector ΔU that minimizes

$$J = (R_s - Fx(k_i))^T (R_s - Fx(k_i)) - 2\Delta U^T \Phi^T (R_s - Fx(k_i)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U, \tag{2.24}$$

subject to the inequality constraints

$$\begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \Delta U \leq \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix}, \tag{2.25}$$

where the data matrices are

$$M_1 = \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}; \; N_1 = \begin{bmatrix} -U^{min} + C_1 u(k_i - 1) \\ U^{max} - C_1 u(k_i - 1) \end{bmatrix}; \; M_2 = \begin{bmatrix} -I \\ I \end{bmatrix};$$

$$N_2 = \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}; \; M_3 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}; N_3 = \begin{bmatrix} -Y^{min} + Fx(k_i) \\ Y^{max} - Fx(k_i) \end{bmatrix}.$$

The matrix $\Phi^T \Phi + \bar{R}$ is the Hessian matrix and is assumed to be positive definite. Since the cost function J is a quadratic, and the constraints are linear inequalities, the problem of finding an optimal predictive control becomes one of finding an optimal solution to a standard quadratic programming problem. For compactness of expression, we denote (2.25) by

$$M \Delta U \leq \gamma, \tag{2.26}$$

where M is a matrix reflecting the constraints, with its number of rows equal to the number of constraints and number of columns equal to the dimension of ΔU. When the constraints are fully imposed, the number of constraints is equal to $4 \times m \times N_c + 2 \times q \times N_p$, where m is the number of inputs and q is the number of outputs. The total number of constraints is, in general, greater than the dimension of the decision variable ΔU. Because the receding horizon control law implements the first control movement and ignores the rest of the calculated future control signals, a question naturally arises as to whether it is necessary to impose constraints on all future trajectories of both the control signals and system output. This question will be investigated in a later section of this chapter.

2.4 Numerical Solutions Using Quadratic Programming

The standard quadratic programming problem has been extensively studied in the literature (see for example, Luenberger, 1984, Fletcher, 1981, Boyd and Vandenberghe, 2004). Since this is a field of study in its own right, it requires a considerable effort to completely understand the relevant theory and algorithms. The required numerical solution for MPC is often viewed as an obstacle in the application of MPC. However, what we can do here is to understand the essence of quadratic programming so that we can produce the essential computational programs required. The advantage of doing so is that we can access the code if anything goes wrong; we can also write safety 'jacket' software for real-time applications. These aspects are very important in an industrial environment. To be consistent with the literatures of quadratic programming, the decision variable is denoted by x. The objective function J and the constraints are expressed as

$$J = \frac{1}{2}x^T E x + x^T F \tag{2.27}$$

$$Mx \leq \gamma, \tag{2.28}$$

where E, F, M and γ are compatible matrices and vectors in the quadratic programming problem. Without loss of generality, E is assumed to be symmetric and positive definite.

2.4.1 Quadratic Programming for Equality Constraints

The simplest problem of quadratic programming is to find the constrained minimum of a positive definite quadratic function with linear equality constraints. Each linear equality constraint defines a hyperplane. Positive definite quadratic functions have their level surfaces as hyperellipsoids. Intuitively, the constrained minimum is located at the point of tangency between the boundary of the feasible set and the minimizing hyperellipsoid. Further illustration is given by the following example.

Example 2.4. Minimize

$$J = (x_1 - 2)^2 + (x_2 - 2)^2,$$

subject to

$$x_1 + x_2 = 1.$$

Solution. The global minimum, without constraint, is at

$$x_1 = 2; x_2 = 2.$$

The feasible solutions are the combinations of x_1 and x_2 that satisfy the linear equality. From the constraint, the feasible solution x_2 is expressed as

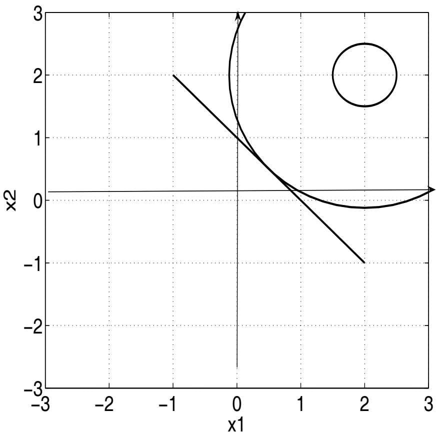


Fig. 2.3. Illustration of constrained optimal solution

$$x_2 = 1 - x_1.$$

Substituting this into the objective function, we obtain

$$\begin{aligned} J &= (x_1 - 2)^2 + (1 - x_1 - 2)^2 \\ &= 2x_1^2 - 2x_1 + 5. \end{aligned} \tag{2.29}$$

In order to minimize J, the derivative $\frac{\partial J}{\partial x_1} = 4x_1 - 2 = 0$, giving the minimizing solution $x_1 = 0.5$. Now, from the constraint equation, $x_2 = 1 - x_1 = 0.5$.

Figure 2.3 illustrates the optimal solution on (x_1,x_2) plane, where the equality constraint defines the straight line and the positive definite quadratic function has its level surface as circles and the constrained minimum is located at the point of tangency between the straight line and the minimizing circle, which is $x_1 = 0.5$ and $x_2 = 0.5$.

The solution of Example 2.4 is easy to understand and it demonstrated the location of the constrained minimum. We consider a general approach to the constrained optimization with equality constraints.

Lagrange Multipliers

To minimize the objective function subject to equality constraints, let us consider the so-called Lagrange expression

$$J = \frac{1}{2}x^T E x + x^T F + \lambda^T (Mx - \gamma). \tag{2.30}$$

It is easy to see that the value of (2.30) subject to the equality constraints $Mx = \gamma$ being satisfied is the same as the original objective function. We now consider (2.30) as an objective function in $n+m$ variables x and λ, where n is the dimension of x and m is the dimension of λ. The procedure of minimization is to take the first partial derivatives with respect to the vectors x and λ, and then equate these derivatives to zero. This gives us the results

$$\frac{\partial J}{\partial x} = Ex + F + M^T\lambda = 0 \tag{2.31}$$

$$\frac{\partial J}{\partial \lambda} = Mx - \gamma = 0. \tag{2.32}$$

The linear equations (2.31) together with (2.32) contain $n+m$ variables x and λ, which are the necessary conditions for minimizing the objective function with equality constraints. The elements of the vector λ are called Lagrange multipliers.

The minimization of the Lagrange expression is straightforward. The optimal λ and x are found via the set of linear equations defined by (2.31) and (2.32) where

$$\lambda = -(ME^{-1}M^T)^{-1}\left(\gamma + ME^{-1}F\right) \tag{2.33}$$

$$x = -E^{-1}(M^T\lambda + F). \tag{2.34}$$

It is interesting to note that (2.34) can be written as two terms:

$$x = -E^{-1}F - E^{-1}M^T\lambda = x^0 - E^{-1}M^T\lambda,$$

where the first term $x^0 = -E^{-1}F$ is the global optimal solution that will give a minimum of the original cost function J without constraints, and the second term is a correction term due to the equality constraints.

The following example is used to illustrate the minimization with equality constraints.

Example 2.5. Minimize

$$J = \frac{1}{2}x^TEx + x^TF,$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 3x_1 - 2x_2 - 3x_3 &= 1, \end{aligned} \tag{2.35}$$

where $E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$; $F = \begin{bmatrix} -2 \\ -3 \\ -1 \end{bmatrix}$.

Solution. Without the equality constraints, the optimal solution is

$$x^0 = -E^{-1}F = \begin{bmatrix} 2 & 3 & 1 \end{bmatrix}^T.$$

Writing the two equality constraints given by (2.35) in matrix form, we obtain the M and γ matrices as

$$M = \begin{bmatrix} 1 & 1 & 1 \\ 3 & -2 & -3 \end{bmatrix}; \ \gamma = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

To use (2.33) the following quantities are required

$$ME^{-1}M^T = \begin{bmatrix} 3 & -2 \\ -2 & 22 \end{bmatrix}; \ ME^{-1}F = \begin{bmatrix} -6 \\ 3 \end{bmatrix}.$$

Note that the determinant $\det(ME^{-1}M^T) = 62$, thus the matrix $ME^{-1}M^T$ is invertible. The λ vector is

$$\lambda = -(ME^{-1}M^T)^{-1} \left(\gamma + ME^{-1}F\right) = \begin{bmatrix} 1.6452 \\ -0.0323 \end{bmatrix}.$$

The x vector that minimizes the objective function is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x^0 - E^{-1}M^T\lambda = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 3 \\ 1 & -2 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} 1.6452 \\ -0.0323 \end{bmatrix} = \begin{bmatrix} 0.4516 \\ 1.2903 \\ -0.7419 \end{bmatrix}.$$

Example 2.6. In this example, we examine what happens to the constrained optimal solution when the linear constraints are dependent. We assume that the objective function

$$J = \frac{1}{2}x^T Ex + x^T F, \tag{2.36}$$

where the matrices $E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $F = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$ and the constraints are

$$\begin{aligned} x_1 + x_2 &= 1 \\ 2x_1 + 2x_2 &= 6. \end{aligned} \tag{2.37}$$

Use graphs to demonstrate that there is no feasible solution of x_1 and x_2 that will satisfy the equality constraints (2.37). In addition, demonstrate that the matrix $M^T E^{-1} M$ is not invertible.

Solution. Figure 2.4 shows that the two equality constraints are defined by two parallel lines on the (x_1, x_2) plane. Because the lines do not intersect, there does not exist a feasible set of parameters x_1 and x_2 that will simultaneously satisfy both linear equations. We can also examine what happens to the Lagrange multipliers when there is no feasible solution. The M and γ matrices are

$$M = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}; \gamma = \begin{bmatrix} 1 \\ 6 \end{bmatrix}.$$

Note that because of the linear dependency in the constraints, the matrix $ME^{-1}M^T = \begin{bmatrix} 2 & 4 \\ 4 & 8 \end{bmatrix}$ has a zero determinant, and does not have an inverse.

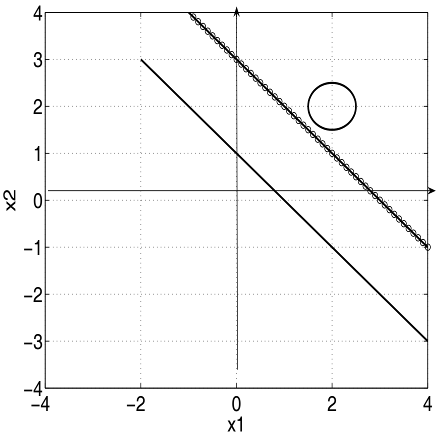


Fig. 2.4. Illustration of no feasible solution of the constrained optimization problem. Solid-line $x_1 + x_2 = 1$; darker-solid-line $2x_1 + 2x_2 = 6$

In summary, in order to find the optimal constrained solution, the linear equality constraints are required to be linearly independent.

Example 2.7. In this example, we will show how the number of equality constraints is also an issue in the constrained optimal solution.

Once again, we use the same objective function as in Example 2.5 where $E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; F = \begin{bmatrix} -2 \\ -3 \\ -1 \end{bmatrix}$, but add an extra constraint to the original constraints so that

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 3x_1 - 2x_2 - 3x_3 &= 1 \\ x_1 - 3x_2 + 2x_3 &= 1. \end{aligned} \tag{2.38}$$

Solution. The additional constraint is independent of the first two original constraints. Now the only feasible solution that satisfies all the constraints is $x = M^{-1}\gamma = \begin{bmatrix} 0.6552 \; 0.069 \; 0.2759 \end{bmatrix}^T$, which is the unique solution of the linear equations (2.38). There is no point in proceeding to the optimization of the objective function, because the only feasible solution is the constrained optimal solution.

In summary, the number of equality constraints is required to be less than or equal to the number of decision variables (*i.e.*, x). If the number of equality constraints equals the number of decision variables, the only feasible solution is the one that satisfies the constraints and there is no additional variable in x that can be used to optimize the original objective function. If the number of equality constraints is greater than the number of decision variables, then there is no feasible solution to satisfy the constraints. Alternatively, the situation is called infeasible.

2.4.2 Minimization with Inequality Constraints

In the minimization with inequality constraints, the number of constraints could be larger than the number of decision variables. The inequality constraints $Mx \leq \gamma$ as in (2.28) may comprise active constraints and inactive constraints. An inequality $M_i x \leq \gamma_i$ is said to be active if $M_i x = \gamma_i$ and inactive if $M_i x < \gamma_i$, where M_i together with γ_i form the *ith* inequality constraint and are the *ith* row of M matrix and the *ith* element of γ vector, respectively. We introduce the Kuhn-Tucker conditions, which define the active and inactive constraints in terms of the Lagrange multipliers.

Kuhn-Tucker Conditions

The necessary conditions for this optimization problem (Kuhn-Tucker conditions) are

$$\begin{aligned} Ex + F + M^T\lambda &= 0 \\ Mx - \gamma &\leq 0 \\ \lambda^T(Mx - \gamma) &= 0 \\ \lambda &\geq 0, \end{aligned} \tag{2.39}$$

where the vector λ contains the Lagrange multipliers. These conditions can be expressed in a simpler form in terms of the set of active constraints. Let S_{act} denote the index set of active constraints. Then the necessary conditions become

$$Ex + F + \sum_{i \in S_{act}} \lambda_i M_i^T = 0$$

$$M_i x - \gamma_i = 0 \quad i \in S_{act} \tag{2.40}$$

$$M_i x - \gamma_i < 0 \quad i \notin S_{act} \tag{2.41}$$

$$\lambda_i \geq 0 \quad i \in S_{act} \tag{2.42}$$

$$\lambda_i = 0 \quad i \notin S_{act}, \tag{2.43}$$

where M_i is the *ith* row of the M matrix. In other words, (2.40) says that for the *ith* row, $M_i x - \gamma_i = 0$ means that this is an equality constraint, hence an active constraint. In contrast, $M_i x - \gamma_i < 0$ (see (2.41)) means that the constraint is satisfied, hence it is an inactive constraint. For an active constraint, the corresponding Lagrange multiplier is non-negative (see (2.42)), whilst the Lagrange multiplier is zero if the constraint is inactive (see (2.43)).

It is clear that if the active set were known, the original problem could be replaced by the corresponding problem having equality constraints only.

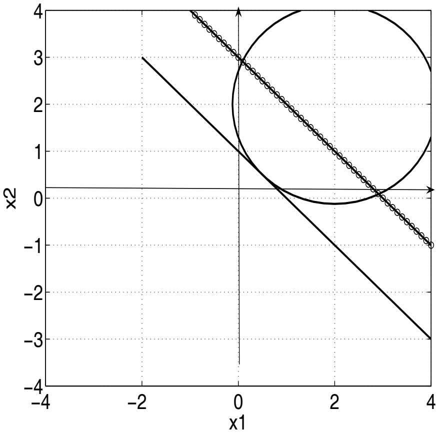


Fig. 2.5. Illustration of the constrained optimization problem with inequality constraints. Solid-line $x_1 + x_2 = 1$; darker-solid-line $3x_1 + 3x_2 = 6$

Explicitly, supposing that M_{act} and λ_{act} are given, the optimal solution with inequality solution has the closed-form

$$\lambda_{act} = -(M_{act}E^{-1}M_{act}^T)^{-1}(\gamma_{act} + M_{act}E^{-1}F) \quad (2.44)$$

$$x = -E^{-1}(F + M_{act}^T\lambda_{act}). \quad (2.45)$$

Example 2.8. In Example 2.6, we showed that when optimizing with equality constraints, if the constraints are dependent, then there is no feasible solution. In this example, we examine what happens to the optimal solution if they are inequality constraints. We assume that the objective function

$$J = \frac{1}{2}x^T Ex + x^T F, \quad (2.46)$$

where the matrices $E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $F = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$ and the constraints are

$$x_1 + x_2 \leq 1 \quad (2.47)$$

$$2x_1 + 2x_2 \leq 6. \quad (2.48)$$

Solution. Clearly the set of variables that satisfy inequality (2.47) will also satisfy the inequality (2.48). This is illustrated in Figure 2.5. Thus, the constraint (2.47) is an active constraint, while the constraint (2.48) is an inactive constraint. We find the constrained optimum by minimizing J subject to equality constraint: $x_1 + x_2 = 1$, which is $x_1 = 0.5$ and $x_2 = 0.5$. We verify that indeed with this set of x_1 and x_2 values, inequality (2.48) is satisfied.

Active Set Methods

The idea of active set methods is to define at each step of an algorithm a set of constraints, termed the working set, that is to be treated as the active set. The working set is chosen to be a subset of the constraints that are actually active at the current point, and hence the current point is feasible for the working set. The algorithm then proceeds to move on the surface defined by the working set of constraints to an improved point. At each step of the active set method, an equality constraint problem is solved. If all the Lagrange multipliers $\lambda_i \geq 0$, then the point is a local solution to the original problem. If, on the other hand, there exists a $\lambda_i < 0$, then the objective function value can be decreased by relaxing the constraint i (*i.e.*, deleting it from the constraint equation). During the course of minimization, it is necessary to monitor the values of the other constraints to be sure that they are not violated, since all points defined by the algorithm must be feasible. It often happens that while moving on the working surface, a new constraint boundary is encountered. It is necessary to add this constraint to the working set, then proceed to the re-defined working surface. To illustrate the basic idea of active set methods, let us look at the example below.

Example 2.9. Optimize the objective function where

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; F = \begin{bmatrix} -2 \\ -3 \\ -1 \end{bmatrix},$$

subject to the inequality constraints:

$$\begin{aligned} x_1 + x_2 + x_3 &\leq 1 \\ 3x_1 - 2x_2 - 3x_3 &\leq 1 \\ x_1 - 3x_2 + 2x_3 &\leq 1. \end{aligned} \tag{2.49}$$

Solution. The feasible solution of equality constraints (2.49) exists, which is the solution of the linear equations

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 3x_1 - 2x_2 - 3x_3 &= 1 \\ x_1 - 3x_2 + 2x_3 &= 1. \end{aligned} \tag{2.50}$$

Thus, the three equality constraints are taken as the first working set. We calculate the Lagrange multiplier for the three constraints leading to

$$\lambda = -(ME^{-1}M^T)^{-1}\left(\gamma + ME^{-1}F\right) = \begin{bmatrix} 1.6873 \\ 0.0309 \\ -0.4352 \end{bmatrix}.$$

Clearly the third element in λ is negative, therefore, the third constraint is an inactive constraint and will be dropped from the constrained equation set. We take the first two constraints as the active constraints, and solve the optimization problem as minimizing

$$J = \frac{1}{2}x^T Ex + x^T F,$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \\ 3x_1 - 2x_2 - 3x_3 &= 1, \end{aligned} \tag{2.51}$$

which is the identical problem to the one given in Example 2.5. We solve this equality constraint problem, obtaining as before

$$\lambda = \begin{bmatrix} 1.6452 \\ -0.0323 \end{bmatrix}.$$

Clearly the second element in λ is negative. We drop the second constraint and solve the optimization problem as

$$J = \frac{1}{2}x^T Ex + x^T F,$$

subject to

$$x_1 + x_2 + x_3 = 1.$$

Once more, we solve this equality constrained optimization problem, and obtain $\lambda = \frac{5}{3}$, leading to $x = \begin{bmatrix} 0.3333 \; 1.3333 \; -0.6667 \end{bmatrix}^T$. Clearly, the optimal solution x satisfies the equality constraint. We also check whether the rest of the inequality constraints (2.49) are satisfied. They are all indeed satisfied.

There are several comments as follows.

1. In the case of equality constraints, the maximum number of equality constraints equals the number of decision variables. In this example, it is 3, and the only feasible solution x is to satisfy the equality constraints (see (2.50)). In contrast, in the case of inequality constraints, the number of inequality constraints is permitted to be larger than the number of decision variables, as long as they are not all active. In this example, only one constraint becomes active so it becomes an equality constraint. Once the optimal solution is found against this active constraint, the rest of the inequalities are automatically satisfied.
2. It is clear that an iterative procedure is required to solve the optimization problem with inequality constraints, because we did not know which constraints would become active constraints. If the active set could be identified in advance, then the iterative procedure would be shortened.
3. Note that the conditions for the inequality constraints are more relaxed than the case of imposing equality constraints. For instance, the number of constraints is permitted to be greater than the number of decision

variables, and the set of inequality constraints is permitted to be linearly dependent. However, these relaxations are only permitted to the point that the active constraints need to be linearly independent and the number of active constraints needs to be less than or equal to the number of decision variables.

2.4.3 Primal-Dual Method

The family of active methods belongs to the group of primal methods, where the solutions are based on the decision variables (also called primal variables in the literature). In the active set methods, the active constraints need to be identified along with the optimal decision variables. If there are many constraints, the computational load is quite large. Also, the programming of an active method is not a straightforward task, as we illustrated through Example 2.9. A dual method can be used systematically to identify the constraints that are not active. They can then be eliminated in the solution. The Lagrange multipliers are called dual variables in the optimization literature. This method will lead to very simple programming procedures for finding optimal solutions of constrained minimization problems.

The dual problem to the original primal problem is derived as follows. Assuming feasibility (*i.e.*, there is an x such that $Mx < \gamma$), the primal problem is equivalent to

$$\max_{\lambda \geq 0} \min_{x} [\frac{1}{2} x^T E x + x^T F + \lambda^T (Mx - \gamma)]. \tag{2.52}$$

The minimization over x is unconstrained and is attained by

$$x = -E^{-1}(F + M^T \lambda). \tag{2.53}$$

Substituting this in (2.52), the dual problem is written as

$$\max_{\lambda \geq 0} (-\frac{1}{2} \lambda^T H \lambda - \lambda^T K - \frac{1}{2} F^T E^{-1} F), \tag{2.54}$$

where the matrices H and K are given by

$$H = ME^{-1}M^T \tag{2.55}$$
$$K = \gamma + ME^{-1}F. \tag{2.56}$$

Thus, the dual is also a quadratic programming problem with λ as the decision variable. Equation (2.54) is equivalent to

$$\min_{\lambda \geq 0} (\frac{1}{2} \lambda^T H \lambda + \lambda^T K + \frac{1}{2} \gamma^T E^{-1} \gamma). \tag{2.57}$$

Note that the dual problem may be much easier to solve than the primal problem because the constraints are simpler (see Hildreth's quadratic programming procedure in the next section).

The set of optimal Lagrange multipliers that minimize the dual objective function

$$J = \frac{1}{2}\lambda^T H\lambda + \lambda^T K + \frac{1}{2}\gamma^T E^{-1}\gamma, \tag{2.58}$$

subject to $\lambda \geq 0$, are denoted as λ_{act}, and the corresponding constraints are described by M_{act} and γ_{act}. With the values of λ_{act} and M_{act}, the primal variable vector x is obtained using

$$x = -E^{-1}F - E^{-1}M_{act}^T\lambda_{act}, \tag{2.59}$$

where the constraints are treated as equality constraints in the computation.

2.4.4 Hildreth's Quadratic Programming Procedure

A simple algorithm, called Hildreth's quadratic programming procedure (Luenberger, 1969, Wismer and Chattergy, 1978), was proposed for solving this dual problem. In this algorithm, the direction vectors were selected to be equal to the basis vectors $e_i = [0\ 0\ \ldots\ 1\ \ldots\ 0\ 0]^T$. Then, the λ vector can be varied one component at a time. At a given step in the process, having obtained a vector $\lambda \geq 0$, we fix our attention on a single component λ_i. The objective function may be regarded as a quadratic function in this single component. We adjust λ_i to minimize the objective function. If that requires $\lambda_i < 0$, we set $\lambda_i = 0$. In either case, the objective function is decreased. Then, we consider the next component λ_{i+1}. If we consider one complete cycle through the components to be one iteration taking the vector λ^m to λ^{m+1}, the method can be expressed explicitly as

$$\lambda_i^{m+1} = max(0, w_i^{m+1}), \tag{2.60}$$

with

$$w_i^{m+1} = -\frac{1}{h_{ii}}[k_i + \sum_{j=1}^{i-1} h_{ij}\lambda_j^{m+1} + \sum_{j=i+1}^{n} h_{ij}\lambda_j^m], \tag{2.61}$$

where the scalar h_{ij} is the $ijth$ element in the matrix $H = ME^{-1}M^T$, and k_i is the ith element in the vector $K = \gamma + ME^{-1}F$. Also note that in (2.61) there are two sets of λ values in the computation: one involves λ^m and one involves the updated λ^{m+1}.

Because the converged λ^* vector contains either zero or positive values of the Lagrange multipliers, by (2.53), we have

$$x = -E^{-1}(F + M^T\lambda^*). \tag{2.62}$$

There are a few comments to be made. First, Hildreth's quadratic programming algorithm is based on an element-by-element search, therefore, it does not require any matrix inversion. As a result, if the active constraints are linearly independent and their number is less than or equal to the number of decision variables, then the dual variables will converge. However, if one or both of these requirements are violated, then the dual variables will not converge to a set of fixed values. The iteration will terminate when the iterative counter reaches its maximum value. Because there is no matrix inversion, the computation will continue without interruption. As we will observe in the coming examples, the algorithm will give a compromised, near-optimal solution with constraints if the situation of conflict constraints arises. This is one of the key strengths of using this approach in real-time applications, because the algorithm's ability to automatically recover from an ill-conditioned constrained problem is paramount for the safety of plant operation.

When the conditions are satisfied, the one-dimensional search technique in Hildreth's quadratic programming procedure has been shown to converge to the set of λ^*, where λ^* contains zeros for inactive constraints and the positive components corresponding to the active constraints. The positive component collected as a vector is called λ^*_{act} with its value defined by

$$\lambda^*_{act} = -(M_{act}E^{-1}M^T_{act})^{-1}(\gamma_{act} + M_{act}E^{-1}F), \tag{2.63}$$

where M_{act} and γ_{act} are the constraint data matrix and vector with the deletion of the row elements that corresponding to the zero elements in λ^*. The proof of the convergence relies on the existence of a set of bounded λ^*_{act}. This is virtually determined by the existence of the $(M_{act}E^{-1}M^T_{act})^{-1}$ (see Wismer and Chattergy (1978)).

Example 2.10. Minimize the cost function:

$$J = \frac{1}{2}x^T E x + F^T x,$$

where $E = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$; $F = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$. The constraints are $0 \le x_1$, $0 \le x_2$ and

$$3x_1 + 2x_2 \le 4.$$

Solution. We form the linear inequality constraints

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}. \tag{2.64}$$

The global optimal solution without constraints is

$$\begin{bmatrix} x_1^0 \\ x_2^0 \end{bmatrix} = -E^{-1}F = -\begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

By substituting the global optimal solution into (2.64), we note that the inequality constraints are violated, with respect to the third constraint (*i.e.*, $3 + 2 > 4$).

To find the optimal λ^*, we form

$$H = \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -1 & 0 & 3 \\ 0 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -5 \\ 1 & 2 & -7 \\ -5 & -7 & 29 \end{bmatrix} \tag{2.65}$$

$$K = \gamma + ME^{-1}F = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}. \tag{2.66}$$

The iteration starts at $k = 0$, with the initial conditions of $\lambda_1^0 = \lambda_2^0 = \lambda_3^0 = 0$.
At $k = 1$,

$$w_1^1 + 1 = 0 \tag{2.67}$$
$$\lambda_1^1 + 2w_2^1 + 1 = 0 \tag{2.68}$$
$$-5\lambda_1^1 - 7\lambda_2^1 + 29w_3^1 - 1 = 0. \tag{2.69}$$

Solving (2.67) gives $\lambda_1^1 = max(0, w_1^1) = 0$, solving (2.68) gives $\lambda_2^1 = max(0, w_2^1) = 0$ and solving (2.69) gives $\lambda_3^1 = max(0, w_3^1) = 0.0345$.
At $k = 2$,

$$w_1^2 + \lambda_2^1 - 5\lambda_3^1 + 1 = 0 \tag{2.70}$$
$$\lambda_1^2 + 2w_2^2 - 7\lambda_3^1 + 1 = 0 \tag{2.71}$$
$$-5\lambda_1^2 - 7\lambda_2^2 + 29w_3^2 - 1 = 0. \tag{2.72}$$

This gives $\lambda_1^2 = max(0, w_1^2) = 0$, $\lambda_2^2 = max(0, w_2^2) = 0$ and $\lambda_3^2 = max(0, w_3^2) = 0.0345$. Since $\lambda^2 = \lambda^1$, the iterative procedure has converged. The optimal solution of λ is $\lambda_1^* = 0$, $\lambda_2^* = 0$ and $\lambda_3^* = 0.0345$. The optimal solution of x is given by

$$x^* = \begin{bmatrix} x_1^0 \\ x_2^0 \end{bmatrix} - E^{-1}M^T\lambda^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.1724 \\ 0.2414 \end{bmatrix} = \begin{bmatrix} 0.8276 \\ 0.7586 \end{bmatrix}. \tag{2.73}$$

From (2.73), it is seen that the constrained optimal solution consists of two parts. One is identical to the global optimal solution, and the second part is a correction term due to the active constraint.

Example 2.11. Solve a quadratic programming problem where the constraints are defined by $0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$ and the objective function is defined by

$$J = \frac{1}{2}[(x_1 - 2)^2 + (x_2 - 2)^2]. \tag{2.74}$$

Solution. The inequalities can be written as

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{2.75}$$

The objective function can be written as

$$J = \frac{1}{2}x^T E x + F^T x, \tag{2.76}$$

where $E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $F = \begin{bmatrix} -2 & -2 \end{bmatrix}^T$. The global optimal solution is

$$\begin{bmatrix} x_1^0 \\ x_2^0 \end{bmatrix} = -E^{-1}F = \begin{bmatrix} 2 \\ 2 \end{bmatrix}. \tag{2.77}$$

The upper limits for x_1 and x_2 are violated. To follow Hildreth's quadratic programming procedure, we define

$$\begin{aligned} H = ME^{-1}M^T &= \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \end{aligned} \tag{2.78}$$

$$K = \gamma + ME^{-1}F \tag{2.79}$$

$$= \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -2 \\ -2 \end{bmatrix} = \begin{bmatrix} -1 \\ 2 \\ -1 \\ 2 \end{bmatrix}. \tag{2.80}$$

We start the one-dimensional search with $\lambda_1^0 = \lambda_2^0 = \lambda_3^0 = \lambda_4^0 = 0$.
At $k = 1$,

$$w_1^1 - 1 = 0 \tag{2.81}$$

$$-\lambda_1^1 + w_2^1 + 2 = 0 \tag{2.82}$$

$$w_3^1 - 1 = 0 \tag{2.83}$$

$$-\lambda_3^1 + w_4^1 + 2 = 0. \tag{2.84}$$

This gives $\lambda_1^1 = max(0, w_1^1) = 1$, $\lambda_2^1 = max(0, w_2^1) = 0$, $\lambda_3^1 = max(0, w_3^1) = 1$, $\lambda_4^1 = max(0, w_4^1) = 0$.
At $k = 2$,

$$w_1^2 - \lambda_2^1 - 1 = 0 \tag{2.85}$$

$$-\lambda_1^2 + w_2^2 + 2 = 0 \tag{2.86}$$

$$w_3^2 - \lambda_4^1 - 1 = 0 \tag{2.87}$$

$$-\lambda_3^2 + w_4^2 + 2 = 0. \tag{2.88}$$

Similarly, solving (2.85) gives $\lambda_1^2 = max(0, w_1^2) = 1$, solving (2.86) gives $\lambda_2^2 = max(0, w_2^2) = 0$, solving (2.87) gives $\lambda_3^2 = max(0, w_3^2) = 1$, and $\lambda_4^2 = max(0, w_4^2) = 0$. Since $\lambda^2 = \lambda^1$, the iterative procedure has converged. The optimal solution of λ is $\lambda_1^* = 1$, $\lambda_2^* = 0$, $\lambda_3^* = 1$ and $\lambda_4^* = 0$.

We delete the inactive constraints and find that the constrained optimal solution is

$$x^* = x^0 - E^{-1}\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{2.89}$$

As we can see, the constrained optimal solution is the solution that consists of an original global optimal solution and a correction term due to the active constraints.

2.4.5 MATLAB Tutorial: Hildreth's Quadratic Programming

Tutorial 2.1. *The objective of this tutorial is to demonstrate how to solve the constrained optimization problem using Hildreth programming. The problem is written as minimizing*

$$J = \frac{1}{2}\eta^T H\eta + \eta^T f, \tag{2.90}$$

subject to constraints

$$A_{cons}\eta \leq b. \tag{2.91}$$

Step by Step

1. *Create a new file called QPhild.m.*
2. *The program finds the global optimal solution and checks if all the constraints are satisfied. If so, the program returns the optimal solution η. If not, the program then begins to calculate the dual variable λ.*
3. *Enter the following program into the file:*

```
function eta=QPhild(H,f,A_cons,b);
% E=H;
% F=f;
% M=A_cons;
% gamma=b;
% eta =x
[n1,m1]=size(A_cons);
eta=-H\f;
kk=0;
for i=1:n1
if (A_cons(i,:)*eta>b(i)) kk=kk+1;
 else
kk=kk+0;
 end
end
if (kk==0) return; end
```

4. *Note that in the quadratic programming procedure, the ith Lagrange multiplier λ_i becomes zero if the corresponding constraint is not active. Otherwise it is positive. We need to calculate the Lagrange multipliers iteratively. We will first set-up the matrices of the dual quadratic programming, followed by the computation of the Lagrange multipliers.*
5. *Continue entering the following program into the file:*

```
P=A_cons*(H\A_cons');
d=(A_cons*(H\f)+b);
[n,m]=size(d);
x_ini=zeros(n,m);
lambda=x_ini;
al=10;
for km=1:38
%find the elements in the solution vector one by one
% km could be larger if the Lagranger multiplier has a slow
% convergence rate.
lambda_p=lambda;
for i=1:n
w= P(i,:)*lambda-P(i,i)*lambda(i,1);
w=w+d(i,1);
la=-w/P(i,i);
lambda(i,1)=max(0,la);
end
al=(lambda-lambda_p)'*(lambda-lambda_p);
if (al<10e-8); break; end
end
```

6. *We can directly use the λ vector and the constraint equation to calculate the changes in η due to the active constraints, because the elements in λ are either positive or zero.*
7. *Continue entering the following program into the file:*

```
eta=-H\f -H\A_cons'*lambda;
```

8. *Test your program using the data matrices generated from Examples 2.10 and 2.11. If the constrained optimal solutions are identical to the solutions given in the examples, then your program is correct.*

2.4.6 Closed-form Solution of $\boldsymbol{\lambda}^*$

We noticed that the Hildreth quadratic programming procedure produces the optimal λ^* that has zeros and the components corresponding to the active constraints. The converged vector is called λ^*_{act} with its value defined by

$$\lambda^*_{act} = -(M_{act}E^{-1}M^T_{act})^{-1}(\gamma_{act} + M_{act}E^{-1}F), \tag{2.92}$$

where M_{act} and γ_{act} are the constraint data matrix and vector with the deletion of the row elements that correspond to the zero elements in λ^*. Thus, if we could correctly identify *a priori* the active constraints, then we can compute the closed-form solution of the constrained optimization problem.

Example 2.12. Suppose that with *a priori* knowledge, the third constraint in Example 2.10 is known to be active. Find the optimal solution using the closed-form solution.

Solution. With the given information, we let $\lambda_1^* = \lambda_2^* = 0$. We form

$$M_{act} = \begin{bmatrix} 3 & 2 \end{bmatrix}; \gamma_{act} = 4$$

$$\lambda_{act}^* = -(M_{act}E^{-1}M_{act}^T)^{-1}(\gamma_{act} + M_{act}E^{-1}F) = \frac{1}{29} = 0.0345.$$

The results are identical to those obtained in Example 2.10.

Example 2.13. Assume that the active constraints in Example 2.11 are number one and number three. Find the optimal solution with respect to the constraints using the closed-form solution.

Solution. With the *a priori* knowledge, $\lambda_2^* = \lambda_4^* = 0$. The active constraints are formed using number one and number three constraints as

$$M_{act} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \gamma_{act} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Hence, the optimal dual variable is solved as

$$\lambda_{act}^* = -(M_{act}E^{-1}M_{act}^T)^{-1}(\gamma_{act} + M_{act}E^{-1}F) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Because the elements in λ_{act}^* are positive, the solution is accepted as the optimal solution for λ^*. Again, the results are identical to those obtained in Example 2.11.

This technique of guessing active constraints is useful in the situation when the computational speed is critical for an application, with which we can resort to the closed-form solution of a constrained control problem. Also, the computational speed is increased if we only need to search for part of the active constraints set, while the other part comes from guesswork.

2.5 Predictive Control with Constraints on Input Variables

This section will present several worked examples of predictive control showing how constraints on the input variable $u(k)$ are imposed. The constraints include those on rate of change and amplitude constraints. These constraints are commonly encountered in industrial applications. MATLAB tutorials for constrained control can be found in Chapter 3.

2.5.1 Constraints on Rate of Change

We present two worked examples of predictive control here. The first example is to show how to incorporate constraints for the first element in ΔU, and the second example is to show how to incorporate constraints for all the elements in ΔU. The first example demands less in computation, but sometimes it can compromise the closed-loop performance of the predictive control system.

Example 2.14. A continuous-time plant is described by a transfer function model,

$$G(s) = \frac{10}{s^2 + 0.1s + 3}, \tag{2.93}$$

which has a pair of poles at $-0.0500 \pm 1.7313j$. Suppose that the system is sampled with an interval $\Delta = 0.1$. Design a discrete-time model predictive control with control horizon $N_c = 3$, $N_p = 20$, and $\bar{R} = 0.01 \times I$. The limit on the rate of change on the control signal is specified as

$$-1.5 \leq \Delta u(k) \leq 3.$$

For this example, we only consider the case of imposing the constraints on the first element of ΔU.

Solution. By following Tutorial 1.1 we first obtain the discrete-time state-space model, then augment the model with an integrator. With the program presented in Tutorial 1.2, we obtain the objective function:

$$J = \Delta U^T(\Phi^T\Phi + \bar{R})\Delta U - 2\Delta U^T\Phi^T(R_s - Fx(k_i)), \tag{2.94}$$

where

$$\Phi^T\Phi = \begin{bmatrix} 0.1760 & 0.1553 & 0.1361 \\ 0.1553 & 0.1373 & 0.1204 \\ 0.1361 & 0.1204 & 0.1057 \end{bmatrix}; \Phi^T F = \begin{bmatrix} 0.1972 & -0.1758 & 1.4187 \\ 0.1740 & -0.1552 & 1.2220 \\ 0.1522 & -0.1359 & 1.0443 \end{bmatrix},$$

and $R_s = \begin{bmatrix} 1.4187 \\ 1.2220 \\ 1.0443 \end{bmatrix} r(k_i)$. $r(k_i)$ and $\hat{x}(k_i)$ are the set-point signal and the estimated state variable at time k_i, respectively. For simplicity, we assume that the observer poles are selected at $0, 0, 0$. The closed-loop system without constraints has its eigenvalues located at $0.6851, 0.9109 \pm 0.1070j$, and $0, 0, 0$. For a unit set-point change, the closed-loop responses are presented in Figure 2.6.

Based on the procedures presented in Section 2.3, the constraints are translated to the two linear inequalities as

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \end{bmatrix} \leq \begin{bmatrix} 3.0000 \\ 1.5000 \end{bmatrix}. \tag{2.95}$$

To demonstrate how the solution evolves, we illustrate the first two steps in the computational procedure.

1. Assume that the initial observer state is zero, and a set-point signal at time $k_i = 1$, is 1. Then, without constraints, the global optimal solution of ΔU is $[6.1083 \;\; 2.3334 \;\; -0.5861]^T$. Thus, the constraint (2.95) is violated. The problem is solved using Hildreth's quadratic programming to obtain

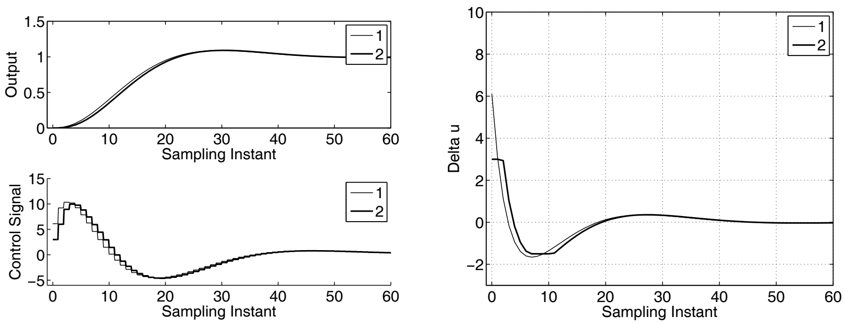
$$\Delta U = [3.0000 \;\; 4.2751 \;\; 1.0494]^T.$$

2. With the first component $\Delta u(1) = 3$, and $y(1) = 0$ the estimated state variable is updated to yield $\hat{x}(2) = [3.0000 \;\; 0 \;\; 0.0015]^T$.
3. Again, the global optimal solution is $\Delta U = [4.6329 \;\; 1.1265 \;\; -1.5559]^T$, which violates the constraint. The optimization problem is again solved using the Hildreth's quadratic programming to obtain

$$\Delta U = [3.0000 \; 2.1466 \;\; -0.6967]^T.$$

4. With updated information on $y(2) = 0.0015$ and $\Delta u(2) = 3$, the estimated state variable is updated with value $\hat{x}(3) = [8.9961 \;\; 3.0000 \;\; 0.0075]^T$.
5. The global optimal solution is $\Delta U = [2.9338 \;\; -0.2126 \;\; -2.5828]^T$, which satisfies the constraint.
6. The computation continues simultaneously in closed-loop simulation and development of constrained control.

Figure 2.6 shows the control signal, plant output and Δu in the presence of the constraints, where the comparison with the unconstrained solution is also illustrated. It is seen from Figure 2.6b that the constraints on Δu are satisfied, whilst Figure 2.6a shows that the control signal and output responses have very small differences in the presence of constraints.



(a) Output and Control Signal (b) Δu

Fig. 2.6. Closed-loop system response with constraints. Key: line (1) without constraints; line (2) with constraints $-1.5 \leq \Delta u(k) \leq 3$

Example 2.15. This example will investigate the scenario where the constraints are imposed for all elements in ΔU, which is the case often referred to in the predictive control literature. The nominal design of predictive control remains the same as in Example 2.14.

Solution. From Section 2.3, when the constraints are fully imposed on all the components in ΔU, they are translated to the six linear inequalities as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \end{bmatrix} \leq \begin{bmatrix} 3 \\ 3 \\ 3 \\ 1.5 \\ 1.5 \\ 1.5 \end{bmatrix}. \tag{2.96}$$

Since there are three decision variables ($N_c = 3$), the number of active constraints should not be greater than 2 at any given time in order to have an optimal solution. We illustrate the solutions for the first two cycles of the computation.

1. At $k_i = 1$, without constraints, the global optimal solution is $\Delta U = [6.1083 \;\; 2.3334 \;\; -0.5861]^T$. Thus, the constraint (2.96) is violated. The problem is solved using Hildreth's quadratic programming to obtain $\Delta U = [3.0000 \;\; 3.0000 \;\; 2.3731]^T$. In comparison with the results in Example 2.14, the first solution has two active constraints. Namely, the first two elements in ΔU are the results from active constraints.
2. At $k_i = 2$, with the first component $\Delta u(1) = 3$, and $y(1) = 0$ the estimated state variable is updated to yield $\hat{x}(2) = [3.0000 \;\; 0 \;\; 0.0015]^T$.
3. Again, the global optimal solution is $\Delta U = [4.6329 \; 1.1265 \;\; - 1.5559]^T$, which violates the constraint. The optimization problem is again solved using Hildreth's quadratic programming to obtain $\Delta U = [3.0000 \; 2.1466 \; - 0.6967]^T$, where only the first constraint becomes activated, therefore the solution is identical to the solution in the second step of Example 2.14.
4. With updated information on $y(2) = 0.0015$ and $\Delta u(2) = 3$, the estimated state variable is updated with value $\hat{x}(3) = [8.9961 \;\; 3.0000 \;\; 0.0075]^T$.
5. The global optimal solution is $\Delta U = [2.9338 \;\; -0.2126 \;\; -2.5828]^T$, where the third element violates the constraint $-1.5 \leq \Delta u(k)$. The optimization problem is solved using Hildreth's quadratic programming to obtain $\Delta U = [2.4885 \;\; -0.6280 \;\; -1.5000]^T$. The effect of the third constraint becoming activated made the first component drop from 2.9338 (see the previous example) to 2.4885.

Because the constraints are imposed on all the element of ΔU, there are possibilities that non-essential constraints become activated. Figure 2.7 shows that this is indeed the case, which results in the non-smooth solution for $\Delta u(k_i)$. In Figure 2.8, we compare the results with those from Example 2.14 where the constraints were only imposed on the first sample of ΔU. The

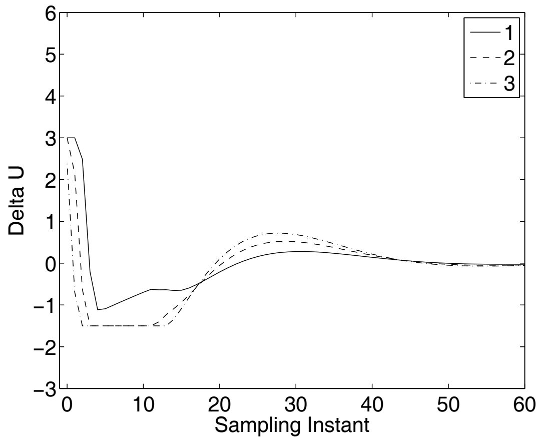
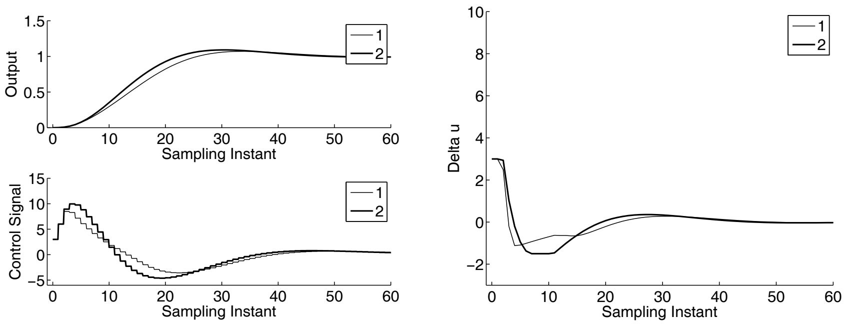


Fig. 2.7. All control elements with constraints. Key: line (1) $\Delta u(k_i)$; line (2) $\Delta u(k_i+1)$; line (3) $\Delta u(k_i+2)$



(a) Output and Control Signal

(b) Δu

Fig. 2.8. Comparison results between Example 2.15 and 2.14. Key: line (1) solution from Example 2.15; line (2) the solution from Example 2.14

difference is caused by the constraints from the set of non-essential constraints becoming active.

2.5.2 Constraints on Amplitude of the Control

The amplitude of the control signal $u(k)$ is another important object for imposing constraint. The examples below illustrate how to impose constraints on the amplitude of the control signal. We will consider two examples. The first example is to impose the constraints on the first sample of the control signal and the second example is to impose constraints on all elements of the control signal.

Example 2.16. We will consider the same system given in Example 2.14 with identical design specification, except that the constraints are changed to

$$-3 \le u(k) \le 6.$$

Again, in this example, we will consider the case of imposing the constraints on the first sample of control.

Solution. Note that at sample time k_i,

$$u(k_i) = \Delta u(k_i) + u(k_i - 1);\ \Delta u(k_i) = [1\ \ 0\ \ 0]\Delta U.$$

Here, ΔU is the parameter vector to be optimized. Therefore, from Section 2.3, the inequality constraints are translated into

$$-3 \le [1\ \ 0\ \ 0]\Delta U + u(k_i - 1) \le 6.$$

That is,

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \end{bmatrix} \le \begin{bmatrix} 6 - u(k_i - 1) \\ 3 + u(k_i - 1) \end{bmatrix}. \tag{2.97}$$

Note that the value of past control signal $u(k_i - 1)$ is embedded into the constraint equations, thus the constraint equations need to be updated as the control system is implemented in real time.

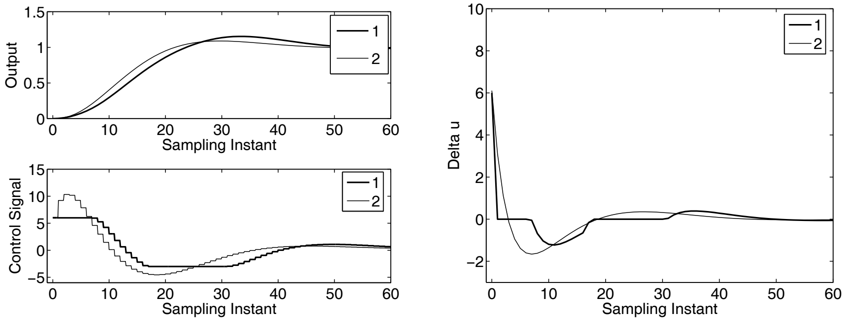
Let us examine the first two cycles of the implementation.

1. When $k_i = 1$, without constraints, the global optimal solution is:

   $$\Delta U = [6.1083\ \ 2.3334\ \ -0.5861]^T,$$

   which gives $u(k_i) = 6.108$, where $u(k_i - 1) = 0$ is assumed as the initial condition. Therefore, the constraint is violated. The quadratic program procedure finds the optimal solution as $\Delta U = [6.0000\ \ 2.4010\ \ -0.5291]^T$, which leads to $u(1) = 6$ satisfying the constraint.
2. With the updated information on the estimated state variable and the information $u(1) = 6$, with constraint on the control amplitude, the optimal solution for $k_i = 2$ is $\Delta U = [0\ \ 1.8921\ \ -0.8643]^T$, which gives the optimal control as $u(2) = 6$ satisfying the constraint.
3. As time progresses, the predictive control system finds the optimal control at each sampling period with respect to the specified constraints on the control signal.

Figure 2.9 shows the constrained control response in comparison with the case without using constraints. It is seen that constraints are satisfied, and the output response remains close to the response obtained from the unconstrained case.



(a) Output and Control Signal (b) Δu

Fig. 2.9. Closed-loop response with constraints (Example 2.16). Key: line (1) solution with constraints; line (2) solution without constraints

Example 2.17. In this example, we will continue the Example 2.16 by considering the case $-3 \leq u(k) \leq 6$, however, where the constraints are imposed on all elements of the control signal.

Solution. Note that

$$u(k_i+1) = \Delta u(k_i+1) + u(k_i) = \Delta u(k_i+1) + \Delta u(k_i) + u(k_i-1);$$

$$u(k_i+2) = \Delta u(k_i+2) + \Delta u(k_i+1) + \Delta u(k_i) + u(k_i-1).$$

In matrix vector form, the expression for the three elements of control in terms of ΔU is

$$\begin{bmatrix} u(k_i) \\ u(k_i+1) \\ u(k_i+2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \end{bmatrix} + \begin{bmatrix} u(k_i-1) \\ u(k_i-1) \\ u(k_i-1) \end{bmatrix}. \tag{2.98}$$

Now, with constraints to be imposed on both upper and lower limits of the control signals, the linear inequalities are formulated into the matrix vector form

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \end{bmatrix} \leq \begin{bmatrix} 6 - u(k_i-1) \\ 6 - u(k_i-1) \\ 6 - u(k_i-1) \\ 3 + u(k_i-1) \\ 3 + u(k_i-1) \\ 3 + u(k_i-1) \end{bmatrix}. \tag{2.99}$$

With the same implementation as before, the predictive control finds the optimal solution that satisfies the six inequality constraints. Figure 2.10 shows the three components in the optimal solution ΔU. It is seen that the two additional constraints on $u(k_i+1)$ and $u(k_i+2)$ are activated during the process of optimization, which is indicated by the zero values of the components in

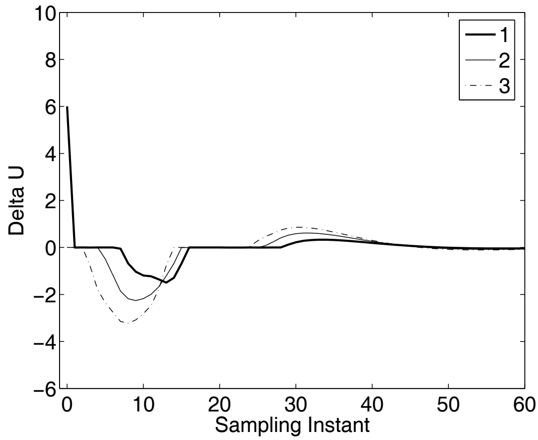
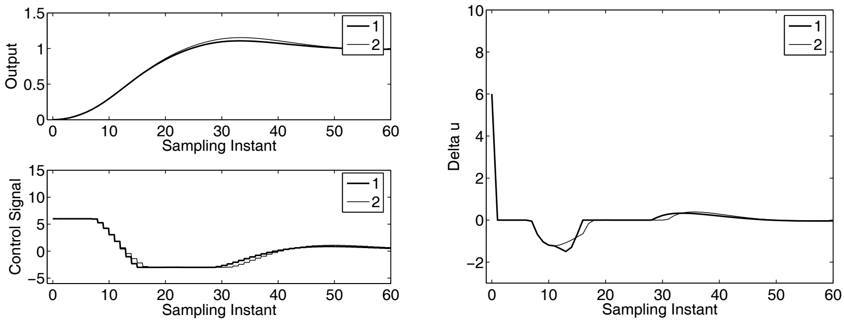


Fig. 2.10. All control elements. Key: line (1) $\Delta u(k_i)$; Line(2) $\Delta u(k_i+1)$; line (3) $\Delta u(k_i+2)$.

ΔU. Strictly speaking, the active constraints are indicated by the positive values of the Lagrange multipliers. However, when the amplitude of the control reaches its limit, the corresponding component in ΔU is zero so as to satisfy the constraint in the solution. In order to illustrate the difference between the approach where the constraints are imposed on all future control and the one with only constraints on $u(k_i)$, Figure 2.11 shows the plant output, control signal and $\Delta u(k)$. Again, the difference is insignificant. However, the number of constraints is three times larger when the constraints are imposed for all future control movements.



(a) Output and Control Signal (b) Δu

Fig. 2.11. Comparison of results between Example 2.16 and 2.17. Key: line (1) solution with constraints on all future control; line (2) solution with constraints on the first sample of future control

2.5.3 Constraints on Amplitude and Rate of Change

It is also a common practice that constraints are imposed on both the amplitude and the rate of change of control signal. If this is required in the design specification, both constraints will be combined together to form a larger set of linear inequalities. Two examples are given below to illustrate the design and implementation procedure. Again, we will consider first the case where constraints are imposed at the sampling instant k_i, and secondly extend the constraints to all future control movements.

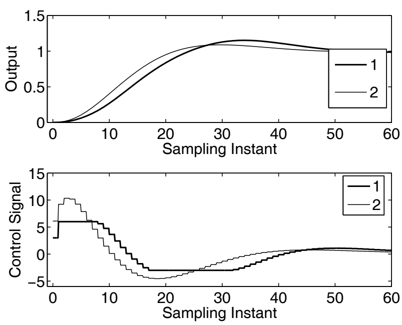
Example 2.18. We consider the same system as in Example 2.14 with constraints on

$$-1.5 \leq \Delta u(k) \leq 3; \quad -3 \leq u(k) \leq 6.$$

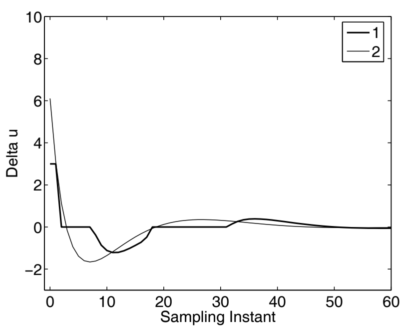
Solution. Assuming that at the sampling instant k_i the constraints are only imposed on $\Delta u(k_i)$ and $u(k_i)$, the set of linear inequality constraints is formulated into the matrix vector form

$$\begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \end{bmatrix} \leq \begin{bmatrix} 3.0000 \\ 1.5000 \\ 6 - u(k_i-1) \\ 3 + u(k_i-1) \end{bmatrix}. \tag{2.100}$$

The first two rows are used for the rate constraints and the last two rows are for the amplitude constraints. We solve the constrained optimization problem by minimizing the objective function J subject to the constraints given by (2.100). The control, output and $\Delta u(k)$ signals are shown in Figure 2.12. It is seen that both constraints are satisfied with a small amount of performance change when compared with the case of optimal control without constraints, which can be viewed in Figure 2.12.



(a) Output and Control Signal



(b) Δu

Fig. 2.12. Closed-loop control response with constraints (Example 2.18). Key: line (1) responses with constraints on both amplitude and rate of change; line (2) responses without constraints

Example 2.19. In this example, we consider the case

$$-1.5 \le \Delta u(k) \le 3; \quad -3 \le u(k) \le 6.$$

However, the constraints will be imposed on all future components of ΔU and $u(k_i), u(k_i+1), u(k_i+2)$ and the results will be compared with those presented in Example 2.18.

Solution. With all the constraints imposed on both the rate of change and the amplitude of the control signal, the inequalities are translated into the matrix vector form

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i+1) \\ \Delta u(k_i+2) \end{bmatrix} \le \begin{bmatrix} 3 \\ 3 \\ 3 \\ 1.5 \\ 1.5 \\ 1.5 \\ 6-u(k_i-1) \\ 6-u(k_i-1) \\ 6-u(k_i-1) \\ 3+u(k_i-1) \\ 3+u(k_i-1) \\ 3+u(k_i-1) \end{bmatrix}. \tag{2.101}$$

The predictive control system is solved by optimizing the objective function (2.94) subject to the constraints defined in (2.101). Figure 2.13 shows the optimal solution for the three components in ΔU. It is seen that all constraints are satisfied. With this predictive control, the plant control, output and $\Delta u(k)$ signals are shown in Figure 2.14. In order to see the improvement of this case over the previous case in Example 2.18, the three signals are compared with these obtained when the constraints on the first sample are imposed.

There is some improvement in the performance with a slightly smaller over-shoot in the output response. However, the difference is very small, as viewed in Figure 2.14.

2.5.4 Constraints on the Output Variable

In this section, we investigate the case where the constraints are imposed on the output variable. Here, we will use a state-space model that has the state variables measurable.

Example 2.20. Assume that a discrete-time system is described by the z-transfer function

$$\frac{Y(z)}{U(z)} = \frac{0.0048z + 0.0047}{(z-1)(z-0.9048)},$$

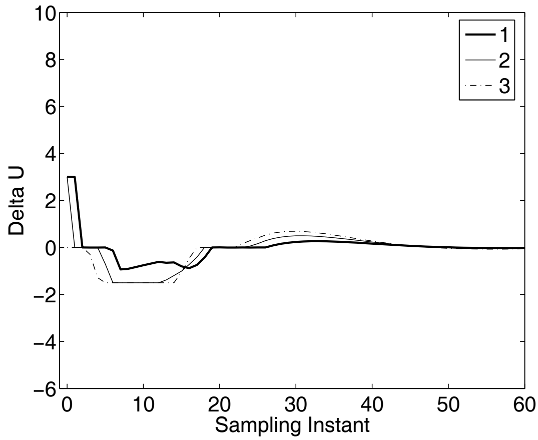
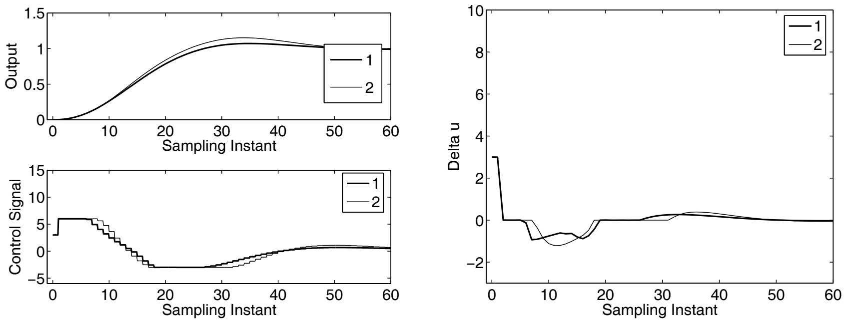


Fig. 2.13. All control elements with constraints. Key: line (1) $\Delta u(k_i)$; line (2) $\Delta u(k_i + 1)$; line (3) $\Delta u(k_i + 2)$



(a) Output and Control Signal

(b) Δu

Fig. 2.14. Comparison of results between Examples 2.18 and 2.19. Key: line (1) solution with constraints on all future control; line (2) solution with constraints on the first sample of future control

which corresponds to the equation with shift operator q,

$$(q^2 - 1.9048q + 0.9048)y(k) = (0.0048q + 0.0047)u(k),$$

where $qy(k) = y(k + 1)$ and $qu(k) = u(k + 1)$. Choosing the state variable $x_m(k) = [y(k)\ y(k-1)\ u(k-1)]^T$, the state-space model with this set of state variables is given by

$$\begin{bmatrix} y(k+1) \\ y(k) \\ u(k) \end{bmatrix} = \begin{bmatrix} 1.9048 & -0.9048 & 0.0047 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ y(k-1) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} 0.0048 \\ 0 \\ 1 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 1 \ 0 \ 0 \end{bmatrix} x_m(k). \tag{2.102}$$

Because the variables in $x_m(k)$ are measurable, there is no need to use an observer in the implementation of the predictive control system.

Assume that $N_p = 46$, $N_c = 8$, and $\bar{R} = 0.1 \times I$. Here, with zero initial conditions of the state variables, a unit step set-point change occurs at sample time $k = 0$. Then at $k = 20$, an input unit step disturbance is introduced into the system and at $k = 40$, a negative unit input disturbance is introduced. The operating constraint is that

$$0 \leq y(k) \leq 1.$$

Design and simulate the predictive control system with output constraints. Also, examine the case when the following input constraints are present,

$$-1.2 \leq u(k) \leq 1.8, \ -0.5 \leq \Delta u(k) \leq 0.5.$$

For simplicity, all constraints are imposed on the first sample of the signals in the optimization window.

Solution. With the performance specification, we design a predictive control system with constraints. Figure 2.15 shows the closed-loop response of the constrained control system. It is seen from the plots that the constraints on the output are satisfied. At sampling time $k = 13$ and $k = 23$, the constraints are active, where we notice two separate sharp drops occurring in the control. The first is due to the slight over-shoot in the set-point change, and the second is due to the input disturbance. By comparing the control signals with and without constraints, we also notice that there are sharp changes on the $\Delta u(k)$ as well as on the control signal $u(k)$ in order to satisfy the constraints on the output. These two sharp changes on both control and increment of control at the same time instant could cause violation of constraints if constraints on the control signal are imposed. We illustrate this by continuing this study with additional constraints on the control.

Now, suppose that there are constraints on the input variables as well as the output variable. With the identical simulation conditions, the constraints on the input variables and output variable are

$$-1.2 \leq u(k) \leq 1.8, \ -0.5 \leq \Delta u(k) \leq 0.5, 0 \leq y(k) \leq 1.$$

Figure 2.16 shows the closed-loop predictive control responses in the presence of the constraints. It is seen from the plots that the output constraints are satisfied. However, the constraints on both the amplitude and increment of the control are violated when the sharp adjustment of control is generated in order to satisfy the constraint on the output. The active constraints on input and output at the same sampling instant become linearly dependent. Therefore, something has to give. Here, without any interferring, Hildreth's programming algorithm chose a solution that satisfies the output constraint and relaxed the input constraints.

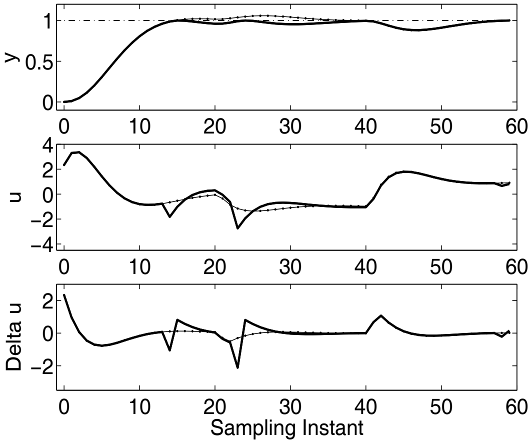


Fig. 2.15. Predictive control with output constraints $0 \leq y(k) \leq 1$. Solid line constrained response; solid dotted line unconstrained response

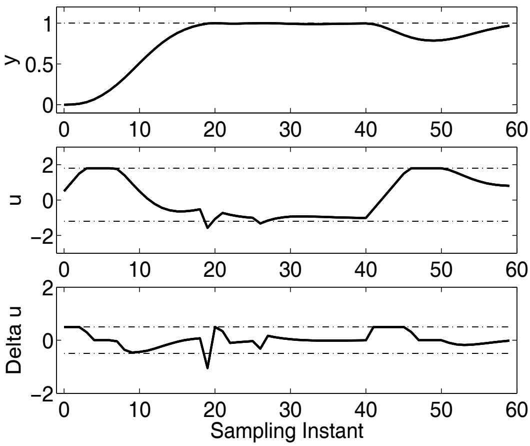


Fig. 2.16. Predictive control with constraints. $-1.2 \leq u(k) \leq 1.8$, $-0.5 \leq \Delta u(k) \leq 0.5$, $0 \leq y(k) \leq 1$

2.6 Summary

This chapter has discussed discrete-time model predictive control with constraints. Imposing constraints in the design and implementation of predictive control system involves the following steps:

1. Defining plant operational limits, including limits on the input variables, the incremental change of the input variables, state variables and plant output variables.

2. Expressing these limits as parameters for the minimum and maximum of u, Δu, x_m and y, with consideration of steady-state information.
3. With parameterization of the future control trajectory, these minimum and maximum values are expressed in the form of inequalities with $\Delta u(k_i)$, $\Delta u(k_i + 1)$, ..., $\Delta u(k_i + N_c - 1)$ as the variables.
4. The design objective of model predictive control becomes the minimization of the original error function subject to the inequality constraints, where the set of parameters $\Delta u(k_i)$, $\Delta u(k_i + 1)$, ..., $\Delta u(k_i + N_c - 1)$ become decision variables.
5. Solving the constrained optimization problem using a quadratic programming procedure at every sampling instance to obtain the optimal solution of the decision variables.

Because the constraints are expressed in terms of inequalities (constraints may or may not be violated at a particular time), in general there is no closed-form solution of the constrained control problem, unless the set of active constraints are known. If the active constraints are known, the optimal solution of the decision variables is expressed in a closed-form. In this chapter, instead of solving the decision variables iteratively, Hildreth's programming procedure was used to identify the active constraints via Lagrange multipliers (or the dual variables). Although it is still a quadratic programming problem on the dual variables, the constraints are much simplified ($\lambda \geq 0$) so a simple iterative procedure was used to obtain the optimal solution of the multipliers. Perhaps even more importantly, the iterative solution does not involve matrix inversion, so in the situation of conflict constraints, the algorithm still delivers a compromised, sub-optimal solution without being numerically unstable. This is particularly important in the real-time implementation of the predictive control system.

There is a rich literature on the topic of predictive control with constraints. The primal methods have dominated the numerical solutions in the classical literature (see for example, Muske and Rawlings, 1993, Ricker, 1985, Zafiriou, 1991) until more recent years specially tailored interior-point methods applicable to MPC have appeared (Rao *et al.*, 1998, Gopal and Biegler, 1998, Hansson, 2000). Some attempts have been made to find analytical solutions (see for example, Bemporad *et al.*, 1999, Seron *et al.*, 2000). A study done by Zheng (1999) has indicated that for stable systems, reducing the number of constraints on the future control movements could cause little performance deterioration. Another interesting approach to the constrained control problem was proposed by Rossiter and Kouvaritakis (1993) where it was solved iteratively using a weighted least squares type of algorithm (Lawson's algorithm). In Tsang and Clarke (1988), optimal solutions were derived for constrained GPC of SISO systems with a control horizon of 1 or 2. The essence of their approach was to take a 'guess' at the active constraints for the two special cases and apply the closed-form solution.

Problems

2.1. Assume that the set of constraints is defined by

$$x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 3, 2x_1 - x_2 \leq 4, x_2 \leq 2,$$

and the objective function is

$$J = \frac{1}{2}x^T Ex + x^T F,$$

where $E = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}$ and $F = \begin{bmatrix} -15 & -7 \end{bmatrix}$.

1. Find the unconstrained minimum of the objective function J.
2. How many constraints are violated with this global optimal solution?
3. Draw the linear inequality constraints on (x_1, x_2) plane, also mark the global optimal solution on the plane. From this plot, take a guess at the active constraints for this constrained optimization problem. Validate your guess by treating the active constraints as equality constraints.
4. If your initial guess is not correct, take another guess at the active constraints until the validation shows that you have correctly found the set of active constraints.
5. What have you learned from this exercise?

2.2. Continue from Problem 2.1. Find the constrained minimum of J by using Hildreth quadratic programming method and compare with the answer you obtained from Problem 2.1.

2.3. Consider the discrete-time DC motor model

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.9048 & 0 \\ 0.0952 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0.0952 \\ 0.0048 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \tag{2.103}$$

Assuming the measurement of shaft position, design a predictive control system that will follow a unit set-point signal with constraints. The closed-loop observer system has poles at 0.1, 0.2 and 0.3, and the control signal satisfies the constraints

$$0 \leq u(k) \leq 0.6; \quad -0.2 \leq \Delta u(k) \leq 0.2.$$

The prediction horizon $N_p = 60$, and control horizon $N_c = 5$; $\bar{R} = I$. The initial conditions of the plant state variable vector and the observer states are assumed to be zero.

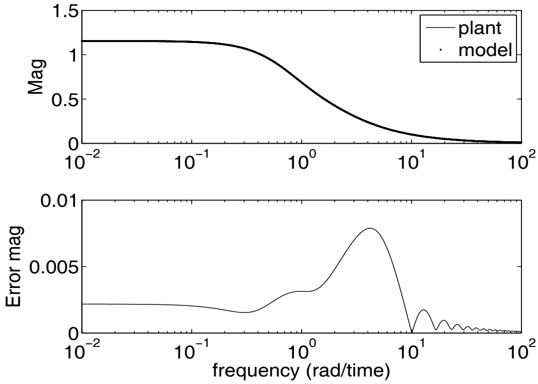


Fig. 2.17. The approximation of a recycle process; comparison of frequency responses

2.4. Transportation of materials in a feedback manner is often called plant with recycle characteristics, which is essentially embedded with time delays in the continuous-time transfer function. An irrational continuous-time transfer function used to describe such a process is given by

$$G_p(s) = \frac{1}{s + 1 - e^{-s-2}}. \tag{2.104}$$

In order to design a feedback control system for this process, one of the approaches is to approximate the continuous-time transfer function with a rational one. Such a continuous-time transfer function approximation (Wang and Cluett, 2000) is found by using a Laguerre model (see Figure 2.17), and has the form

$$G(s) = \frac{1.0117s^2 + 2.1709s + 1.4949}{(s + 1.09)^3}.$$

Assume that the control objective of the predictive control system is to maintain the output to be constant while rejecting input step disturbance. Choosing the sampling interval $\Delta t = 0.2$, $\bar{R} = I$, $N_c = 4$, $N_p = 20$, design a discrete-time model predictive control system with constraints, assuming a unit step set-point signal and zero initial conditions on the state variables, where the constraints are specified as

$$-0.1 \leq \Delta u(k) \leq 0.1; \quad 0 \leq u(k) \leq 1.1.$$

An observer may be used in the control system, where the closed-loop poles of the observer are chosen to be 0.1,0.2, 0.3 and 0.4. The constraints are imposed on the first two samples of the variables. Simulate the nominal closed-loop performance by assuming zero initial conditions of u and y. In this simulation, the plant is assumed to be identical to the approximate model.

http://www.springer.com/978-1-84882-330-3

Model Predictive Control System Design and
Implementation Using MATLAB®
Wang, L.
2009, XXX, 378 p., Hardcover
ISBN: 978-1-84882-330-3