Model Predictive Control with Constraints

We begin this section with a straightforward example:

Example 5.1

A continuous-time model of an undamped oscillator is given by

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

\[
y(t) =
\begin{bmatrix}
0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1(t) \\
x_2(t)
\end{bmatrix}
\]

The eigenvalues of the system matrix are located at \( \lambda_{1,2} = \pm j \frac{2}{\sqrt{10}} \) (purely imaginary), so the natural frequency is \( \omega = \frac{2 \text{ rad/sec}}{\sqrt{10}} \).

Assuming a sampling interval \( \Delta t = 0.1 \), we compute the corresponding discrete-time model:

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

\[
y(k) =
\begin{bmatrix}
0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1(k) \\
x_2(k)
\end{bmatrix}
\]

The eigenvalues of the discrete-time system are located at \( \lambda_{1,2} = 0.9801 \pm j 0.1986 \).

- As we expect, both have modulus 1.0 and lie on the unit circle.
• Our objective is to design a predictive control system so that the output of the plant tracks a unit step reference as fast as possible

• Tuning parameters for the problem are selected as:
  
  \[ N_c = 3 \]
  
  \[ N_p = 10 \]
  
  \[ \bar{R} = 0 \]

⇒ What happens if control magnitude *saturates* at ±20?

• Computing the data matrices for this problem yields,

\[
\Phi^T \Phi = \begin{bmatrix}
6.0067 & 4.8853 & 3.8150 \\
4.8853 & 4.0013 & 3.1475 \\
3.8150 & 3.1475 & 2.4952
\end{bmatrix}
\]

\[
\Phi^T F = \begin{bmatrix}
65.5285 & -25.2099 & -6.1768 \\
53.1281 & -19.6709 & -4.7606 \\
41.3553 & -14.6974 & -3.5334
\end{bmatrix}
\]

\[
\Phi^T R_s = \begin{bmatrix}
-6.1768 \\
-4.7606 \\
-3.5334
\end{bmatrix}
\]

• Computing the state feedback gain matrix we obtain

\[
K_{mpc} = \begin{bmatrix}
17.9064 & -39.0664 & -29.9659
\end{bmatrix}
\]

  – This gives closed-loop eigenvalues at: \[ \begin{bmatrix}
-0.1946, & 0, & 0
\end{bmatrix} \]

• We now design an observer with poles at \[ \begin{bmatrix}
0, & 0, & 0
\end{bmatrix} \] and examine two cases: (i) no constraint on control; and (ii) with constraint on control.
CASE (I) NO CONSTRAINT.

- Assuming the control amplitude has saturation limits at ±20, we write

\[-20 \leq u(k) \leq 20\]

- So, \(u(k) = 20\) when \(u(k) \geq 20\) and \(u(k) = -20\) when \(u(k) \leq 20\)

CASE (II) WITH CONSTRAINT.

- As we might expect, the closed-loop performance deteriorates when these limits are reached
So we ask: is it possible to achieve better performance by incorporating the constraints directly into the model predictive control problem?

Let’s try it...

Formulation of Constrained Control Problems

- The basic idea here is to take the control increment $\Delta u(k)$ obtained for the unconstrained problem and modify it when a constraint becomes active
- This involves formulating an optimization problem directly in the presence of constraints
- Three of the most common types of constraints are placed on:
  - control variable increment
control variable amplitude

output variable

- It is also sometimes desirable to constrain state variables internal to the dynamic process; we'll cover this case as well

Numerical Solutions

- The standard optimization problem for constrained model predictive control can be cast as a *quadratic programming* problem

- For consistency with the prevailing literature, we write the objective function $J$ together with a set of constraint equations as:

$$J = \frac{1}{2} x' E x + x' F$$

$$Mx \leq y$$

where we assume $E$ to be symmetric and positive definite.

The Unconstrained Case

- General optimization problems involve finding a local solution to the problem

$$\min \ J(x), \quad x \in \mathbb{R}^n$$

- Here, $J(x)$ is the objective function and the minimizing point denoted by $x^*$ is found among all $x$ vectors in $\mathbb{R}^n$

- We must first make some assumptions on the existence and uniqueness of $x^*$

  - Clearly $x^*$ may not exist if $J$ is unbounded below
In general, it is only really practical to locate a *local* minimizer, and this may not be a *global* minimizer.

- In this treatment we will assume that first and second derivatives exist, are continuous and are given by:

\[
g(x) = \nabla J(x)
\]

\[
G(x) = \nabla^2 J(x)
\]

- At a local minimizer, the following simple stationarity conditions hold:

\[
g(x^*) = 0
\]

\[
s^t G(x^*)s \geq 0, \quad \forall s
\]

- These are first-order and second-order necessary conditions for a local solution

  - The first condition identifies a *stationary point*
  - The second implies the matrix \( G(x^*) \) is *positive semi-definite* and indicates the presence of non-negative curvature in all directions from the stationary point.
Sufficient conditions for a strict and isolated minimizer $x^*$ are that the above hold and that $G(x^*)$ is positive definite.

- How can we determine numerically whether or not $G(x^*)$ is positive definite?

**Quadratic Programming: Equality Constraints**

- For unconstrained optimization, the necessary conditions above illustrate the requirement for zero slope and non-negative curvature in any direction at $x^*$
- In constrained optimization, there is the additional consideration of a feasible region - i.e., a local minimizer must be a feasible point that satisfies the constraints
- Additionally, there must be no feasible descent directions at $x^*$.
- The simplest problem involves finding 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
- The constrained minimum is then found at the point of tangency between the boundary of the feasible set and the minimizing hyperellipsoid
Example 5.2

Minimize

\[ J = (x_1 - 2)^2 + (x_2 - 2)^2 \]

subject to

\[ x_1 + x_2 = 1 \]

- By inspection we see that the unconstrained minimum occurs at
  \[ x_1 = 2 \]
  \[ x_2 = 2 \]

- Feasible solutions are the combinations of \( x_1 \) and \( x_2 \) that satisfy the linear equality \( x_1 + x_2 = 1 \)

- So we have
  \[ x_2 = 1 - x_1 \]

- Substituting into the expression for \( J \) gives,
  \[ J = (x_1 - 2)^2 + (1 - x_1 - 2)^2 \]
  \[ = 2x_1^2 - 2x_1 + 5 \]

- To minimize \( J \), we satisfy the stationarity condition
  \[ \frac{\partial J}{\partial x_1} = 4x_1 - 2 = 0 \]
  giving,
  \[ x_1 = 0.5 \]
  \[ x_2 = 0.5 \]
**Lagrange Multipliers**

- One approach to minimizing the objective function subject to equality constraints is to augment the original function with the equality constraint multiplied by a *Lagrange multiplier* vector, \( \lambda \):

  \[
  J = \frac{1}{2} x^T E x + x^T F + \lambda^T (M x - y)
  \]

  - This creates a new objective function in the \( n + m \) variables given by the vectors \( x \ (n \times 1) \) and \( \lambda \ (m \times 1) \)

- The solution is found by taking the first partial derivatives with respect to the vectors \( x \) and \( \lambda \) and then equating the derivatives to zero:

  \[
  \frac{\partial J}{\partial x} = E x + F + M^T \lambda = 0
  \]

  \[
  \frac{\partial J}{\partial \lambda} = M x - y = 0
  \]

- From these expressions we can build the *Lagrangian matrix* and write the corresponding linear expression

  \[
  \begin{bmatrix}
  E & M^T \\
  M & 0
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  \lambda
  \end{bmatrix}
  =
  \begin{bmatrix}
  -F \\
  y
  \end{bmatrix}
  
  \]

- If the inverse exists and is expressed as

  \[
  \begin{bmatrix}
  E & M^T \\
  M & 0
  \end{bmatrix}^{-1}
  =
  \begin{bmatrix}
  H & T \\
  T & U
  \end{bmatrix}
  
  \]

  then the solution can be written

  \[
  x^* = -HF + T\lambda
  \]

  \[
  \lambda^* = -T^T F - U\lambda
  \]
Explicit expressions for $H$, $T$ and $U$ (when the inverse exists) are given by

\[
H = E^{-1} - E^{-1} M^T (ME^{-1} M^T)^{-1} ME^{-1}
\]
\[
T = E^{-1} M^T (ME^{-1} M^T)^{-1}
\]
\[
U = (ME^{-1} M^T)^{-1}
\]

This set of equations contains $n + m$ variables in the vectors $x$ and $\lambda$, which form the necessary conditions for minimizing the objective function with equality constraints.

The optimizing $\lambda^*$ and $x^*$ can be found directly from the partial derivatives above where

\[
\lambda^* = -(ME^{-1} M^T)^{-1} (y + ME^{-1} F)
\]
\[
x^* = -E^{-1} (M^T \lambda^* + F)
\]

Note that

\[
x^* = -E^{-1} F - E^{-1} M^T \lambda^* = x^o - E^{-1} M^T \lambda^*
\]

where the first term $x^o$ gives the optimal solution in the absence of constraints, and the second term is a correction factor due to the equality constraint.
Example 5.3

Minimize

\[ J = \frac{1}{2} x' E x + x' F \]

subject to

\[ x_1 + x_2 + x_3 = 1 \]
\[ 3x_1 - 2x_2 - 3x_3 = 1 \]

where

\[
E = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}; \quad F = \begin{bmatrix}
-2 \\
-3 \\
-1 \\
\end{bmatrix}
\]

- The unconstrained optimizing solution is given by

\[
x^o = -E^{-1} F = \begin{bmatrix}
2 \\
3 \\
1 \\
\end{bmatrix}
\]

- Note that the cost at the unconstrained minimum is:

\[
J_o = \frac{1}{2} x_o' E x_o + x_o' F
\]

\[
= \frac{1}{2} \begin{bmatrix}
2 & 3 & 1 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix} \begin{bmatrix}
2 \\
3 \\
1 \\
\end{bmatrix} + \begin{bmatrix}
2 & 3 & 1 \\
\end{bmatrix} \begin{bmatrix}
-2 \\
-3 \\
-1 \\
\end{bmatrix}
\]

\[= \frac{1}{2} \times 14 - 14 = -7 \]

- From the constraint equations, we write,

\[
M = \begin{bmatrix}
1 & 1 & 1 \\
3 & -2 & -3 \\
\end{bmatrix}; \quad \nu = \begin{bmatrix}
1 \\
1 \\
\end{bmatrix}
\]
Computing the minimizing Lagrangian vector $\lambda^*$ gives,

$$\lambda^* = -(ME^{-1}M^T)^{-1}(\gamma + ME^{-1}F)$$

$$= -\begin{bmatrix} 3 & -2 \\ -2 & 22 \end{bmatrix}^{-1} \left( \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -6 \\ 3 \end{bmatrix} \right) = \begin{bmatrix} 1.6452 \\ -0.0323 \end{bmatrix}$$

Hence, $x^*$ is given by,

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x^0 - E^{-1}M^T\lambda^*$$

$$= \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 3 \\ 1 & -2 \\ 1 & -3 \end{bmatrix} \begin{bmatrix} 1.6452 \\ -0.0323 \end{bmatrix}$$

$$= \begin{bmatrix} 0.4516 \\ 1.2903 \\ -0.7419 \end{bmatrix}$$

For the constrained optimizing solution,

$$J^* = (x^*)^t E x^* + (x^*)^t F = -1.6130$$

indicating the cost function cannot achieve the same value as the unconstrained optimum.

Alternatively, building the Lagrangian matrix,

$$\mathcal{L} = \begin{bmatrix} E & M^T \\ M & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 3 \\ 0 & 1 & 0 & 1 & -2 \\ 0 & 0 & 1 & 1 & -3 \\ 1 & 1 & 1 & 0 & 0 \\ 3 & -2 & -3 & 0 & 0 \end{bmatrix}$$
Computing the eigenvalues of $\mathcal{L}$ we have
\[
\lambda\{\mathcal{L}\} = \begin{bmatrix} 5.239 & 2.2441 & 1.000 & -1.2441 & -4.2390 \end{bmatrix}
\]
indicating the matrix is non-singular and invertable.

Solving the linear equation,
\[
\mathcal{L} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} -F \\ \gamma \end{bmatrix}
\]
we again obtain the solution vector
\[
\begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} .4516 \\ 1.2903 \\ -.7419 \\ 1.6452 \\ -.0323 \end{bmatrix}
\]
Example 5.4

Consider once again the objective function,

\[ J = \frac{1}{2} x^t E x + x^t F \]

where,

\[ E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad F = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \]

and the constraints are given by:

\[ x_1 + x_2 = 1 \]
\[ 2x_1 + 2x_2 = 6 \]

- Find the feasible set of solutions
- The plot below shows contours of equal \( J \) for corresponding values of \( x_1 \) and \( x_2 \)
• Note that in general, the number of equality constraints is required to be less than or equal to the number of decision variables.
• What happens when the number of equality constraints exactly equals the number of decision variables?
• What happens when the number of equality constraints is greater than the number of decision variables?

**Quadratic Programming: Inequality Constraints**

• In the previous case, it was necessary that all equality constraints be active at the solution.
• In this case we have to consider the case where the inequality constraints \( Mx \leq \gamma \) may comprise both active and inactive constraints:
  - *active* denotes equality
  - *inactive* denotes strict inequality

**Kuhn-Tucker Conditions**

• The Kuhn-Tucker conditions outline a set of necessary conditions for the optimization problem:

\[
Ex + F + M^T \lambda = 0 \\
Mx - \gamma \leq 0 \\
\lambda^T (Mx - \gamma) = 0 \\
\lambda \geq 0
\]

• In terms of the active constraints, the conditions become:

\[
Ex + F + \sum_{i \in S_{act}} \lambda_i M_i^T = 0
\]
\[ M_i x - \gamma_i = 0 \quad i \in S_{\text{act}} \]
\[ M_i x - \gamma_i < 0 \quad i \notin S_{\text{act}} \]
\[ \lambda_i \geq 0 \quad i \in S_{\text{act}} \]
\[ \lambda_i = 0 \quad i \notin S_{\text{act}} \]

where \( M_i \) is the \( i^{th} \) row of the \( M \) matrix.

- For an active constraint (i.e., \( M_i x - \gamma_i = 0 \)), the corresponding Lagrange multiplier is non-negative.
- Otherwise if the constraint is inactive, the Lagrange multiplier is zero.

\[ \Rightarrow \quad \text{So, for the set of active constraints, the problem looks exactly like the equality constraint optimization!} \]

- Assume the previous problem with the equality constraints replaced with inequality constraints,

\[ J = \frac{1}{2} x^t E x + x^t F \]

where,

\[ E = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad F = \begin{bmatrix} -2 \\ -2 \end{bmatrix} \]

and the inequality constraints are now given by:

\[ x_1 + x_2 \leq 1 \]
\[ 2x_1 + 2x_2 \leq 6 \]

- It is clear that the set of variables that satisfy the first inequality will also satisfy the second.
  - Thus the first of these is an active constraint, while the second is inactive.
We find the constrained optimum by minimizing $J$ subject to the equality constraint $x_1 + x_2 = 1$, which gives $x_1 = 0.5$ and $x_2 = 0.5$, as found previously.

**Active Set Methods**

- The idea here is to construct an algorithm where we define at each step a set of constraints to be treated as the *active set*
  - Chosen to be a subset of the constraints that are active at the current point
  - Current point is *feasible* for the working set
  - Algorithm moves along a surface defined by the working set of constraints to an improved point
  - If all $\lambda_i \geq 0$, the point is a local solution
  - If some $\lambda_i < 0$, then the objective function value can be further decreased by relaxing the constraint
Example 5.5

Optimize the objective function where

\[
E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad F = \begin{bmatrix} -2 \\ -3 \\ -1 \end{bmatrix}
\]

subject to equality constraints

\[
x_1 + x_2 + x_3 \leq 1 \\
3x_1 - 2x_2 - 3x_3 \leq 1 \\
x_1 - 3x_2 + 2x_3 \leq 1
\]

- The feasible solution of the equality constraints exists and is the solution of the linear equations

\[
x_1 + x_2 + x_3 = 1 \\
3x_1 - 2x_2 - 3x_3 = 1 \\
x_1 - 3x_2 + 2x_3 = 1
\]

- The three equality constraints are taken as the first working set

- Calculating the Lagrange multipliers we get

\[
\lambda = -(ME^{-1}M^T)^{-1}(\gamma + ME^{-1}F) = \begin{bmatrix} 1.6873 \\ 0.0309 \\ -0.4352 \end{bmatrix}
\]

- Since the third element is negative, the third constraint is inactive and is dropped from the constrained equation set

- Thus the problem becomes:

\[
J = \frac{1}{2}x^tEx + x^tF
\]
subject to
\[
x_1 + x_2 + x_3 = 1 \\
3x_1 - 2x_2 - 3x_3 = 1
\]

- Solve this equality constraint problem to get

\[
\lambda = \begin{bmatrix}
  1.6452 \\
  -0.0323
\end{bmatrix}
\]

- Here the second element is negative

- The problem collapses once again to

\[
J = \frac{1}{2} x^t E x + x^t F
\]

now subject only to the single constraint

\[
x_1 + x_2 + x_3 = 1
\]

- Solving this problem yields \( \lambda = \frac{5}{3} \), leading to

\[
x^* = \begin{bmatrix}
  0.3333 \\
  1.3333 \\
  -0.6667
\end{bmatrix}
\]

- Clearly, the optimal solution \( x^* \) satisfies the equality constraint

- Checking the others, we obtain

\[
M x^* = \begin{bmatrix}
  1 & 1 & 1 \\
  3 & -2 & -3 \\
  1 & -3 & 2
\end{bmatrix} \begin{bmatrix}
  0.3333 \\
  1.3333 \\
  -0.6667
\end{bmatrix} = \begin{bmatrix}
  1.0 \\
  0.3334 \\
  -5.0
\end{bmatrix} \leq \begin{bmatrix}
  1.0 \\
  1.0 \\
  1.0
\end{bmatrix}
\]

and indeed the rest of the inequality constraints are also satisfied.

- Some observations:
– The maximum number of equality constraints equals the number of decision variables
– The maximum number of inequality constraints can exceed the number of decision variables if they are not all active
– An iterative procedure is required to solve the optimization problem with inequality constraints

**Primal-Dual Methods**

- A *dual* method can be used to identify the constraints that are not active in the optimization problem so that they can be systematically eliminated in the solution
- Assuming feasibility (i.e., there exists an \( x \) such that \( Mx < y \)), the *primal* problem is equivalent to

\[
\max_{\lambda \geq 0} \min_x \left[ \frac{1}{2}x^t Ex + x^t F + \lambda^t (Mx - y) \right]
\]

- The minimum over \( x \) is unconstrained and attained by

\[
x = -E^{-1}(F + M^T \lambda)
\]

- Substituting this into the above expression gives the dual problem

\[
\max_{\lambda \geq 0} \left( -\frac{1}{2} \lambda^t P \lambda - \lambda^t K - \frac{1}{2} F^T E^{-1} F \right)
\]

where

\[
P = ME^{-1}M^T
\]
\[
K = y + ME^{-1}F
\]

- Hence the dual is another quadratic programming problem, only now with \( \lambda \) as the decision variable instead of \( x \)
• Note, the dual problem is equivalent to
\[ \max_{\lambda \geq 0} \left( \frac{1}{2} \lambda^t P \lambda + \lambda^t K + \frac{1}{2} y^T E^{-1} y \right) \]

**Hildreth’s Procedure**

• Hildreth’s quadratic programming procedure is a simple variant of the Gauss-Seidel method for solving a linear set of equations

• Some features:
  - Direction vectors are equal to the standard basis vectors,
    \[ e_i = \left[ \begin{array}{cccccc} 0 & 0 & \ldots & 1 & \ldots & 0 \end{array} \right]^t \]
  - The \( \lambda \) vector is varied one component at a time
  - The objective function can be regarded as a quadratic function in each component
  - Adjust \( \lambda_i \) to minimize the objective function
  - When \( \lambda_i < 0 \) then set \( \lambda_i = 0 \)

• The general method can be expressed as:
  \[ \lambda_i^{m+1} = \max(0, \ w_i^{m+1}) \]
  where,
  \[ w_i^{m+1} = -\frac{1}{p_{ii}}[k_i + \sum_{j=1}^{i-1} p_{ij} \lambda_j^{m+1} + \sum_{j=i+1}^{n} p_{ij} \lambda_j^m] \]

• \( p_{ij} \) is the \( ij^{th} \) element of the matrix \( P = ME^{-1}M^T \)
• \( k_i \) is the \( i^{th} \) element of the vector \( K = y + ME^{-1}F \)
Example 5.6

Minimize the cost function $J = \frac{1}{2} x^T E x + F^T x$ where

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

and the constraints are

$$0 \leq x_1$$
$$0 \leq x_2$$
$$3x_1 + 2x_2 \leq 4$$

- Forming the linear inequality constraint equation, we have

$$M x \leq \gamma$$

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

- The unconstrained optimal solution is found as

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

- Substituting $x_o$ into the expression above shows that the third constraint is violated:

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 5 \end{bmatrix} \not< \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

- Now, find the optimal $\lambda^*$:

$$P = ME^{-1}M^T$$
\[
\begin{bmatrix}
-1 & 0 \\
0 & -1 \\
3 & 2
\end{bmatrix}
\begin{bmatrix}
2 & -1 \\
-1 & 1 \\
-1 & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
-1 & 0 & 3 \\
0 & -1 & 2
\end{bmatrix}
\]
\[
= 
\begin{bmatrix}
1 & 1 & -5 \\
1 & 2 & -7 \\
-5 & -7 & 29
\end{bmatrix}
\]
\[
K = y + ME^{-1}F
\]
\[
= 
\begin{bmatrix}
0 \\
0 \\
4
\end{bmatrix}
+ 
\begin{bmatrix}
-1 & 0 \\
0 & -1 \\
-1 & 1
\end{bmatrix}
\begin{bmatrix}
2 & -1 \\
-1 & 1
\end{bmatrix}^{-1}
\begin{bmatrix}
-1 \\
0
\end{bmatrix}
\]
\[
= 
\begin{bmatrix}
1 \\
1 \\
-1
\end{bmatrix}
\]

- Iteration:

k=0

\[
\lambda_1^0 = \lambda_2^0 = \lambda_3^0 = 0
\]

k=1

\[
w_1^1 + 1 = 0
\]

\[
\lambda_1^1 + 2w_2^1 + 1 = 0
\]

\[
-5\lambda_1^1 - 7\lambda_2^1 + 29w_3^1 - 1 = 0
\]

\[
\lambda_1^1 = \max(0, \ w_1^1) = 0
\]

\[
\lambda_2^1 = \max(0, \ w_2^1) = 0
\]

\[
\lambda_3^1 = \max(0, \ w_3^1) = 0.0345
\]
\( k=2 \)

\[
\begin{align*}
w_1^2 + \lambda_2^1 - 5\lambda_3^1 + 1 &= 0 \\
\lambda_1^2 + 2w_2^2 - 7\lambda_3^1 + 1 &= 0 \\
-5\lambda_1^2 - 7\lambda_2^2 + 29w_3^2 - 1 &= 0
\end{align*}
\]

\[
\begin{align*}
\lambda_1^2 &= \max(0, w_1^2) = 0 \\
\lambda_2^2 &= \max(0, w_2^2) = 0 \\
\lambda_3^2 &= \max(0, w_3^2) = 0.0345
\end{align*}
\]

- We see that the procedure has converged giving the optimal Lagrange vector:

\[
\lambda^* = \begin{bmatrix} 0 \\ 0 \\ 0.0345 \end{bmatrix}
\]

which generates the optimal solution:

\[
x^* = x_o - E^{-1}M^T\lambda^* = \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.1724 \\ 0.2414 \end{bmatrix} = \begin{bmatrix} 0.8276 \\ 0.7586 \end{bmatrix}
\]
Constraints on Rate of Change

Example 5.7

Consider the continuous-time plant described by

\[ G(s) = \frac{10}{s^2 + 0.1s + 3} \]

whose system poles are located at: \(-0.05 \pm j1.7313\).

Assume a sampling interval of \(\Delta T = 0.1\) and design a discrete-time model predictive control system with

\[ N_c = 3 \]
\[ N_p = 20 \]
\[ \bar{R} = 0.01 \times I \]

A constraint is imposed on rate-of-change of the control signal as

\[-1.5 \leq \Delta u(k) \leq 3.0\]

- First, obtain a discrete-time state-space model

  - Continuous-time state-space:

    \[
    A_c = \begin{bmatrix}
    -0.1 & -3 \\
    1 & 0
    \end{bmatrix} \quad B_c = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
    C_c = \begin{bmatrix} 0 & 10 \end{bmatrix} \quad D_c = \begin{bmatrix} 0 \end{bmatrix}
    \]

  - Discrete-time state-space:

    \[
    A_d = \begin{bmatrix}
    0.9752 & -0.2970 \\
    0.0990 & 0.9851
    \end{bmatrix} \quad B_d = \begin{bmatrix} 0.0990 \\ 0.005 \end{bmatrix} \\
    C_d = \begin{bmatrix} 0 & 10 \end{bmatrix} \quad D_d = \begin{bmatrix} 0 \end{bmatrix}
    \]
• Next, form the augmented state-space model:

\[
A = \begin{bmatrix}
.9752 & -.2970 & 0 \\
.0990 & .9851 & 0 \\
.990 & 9.8509 & 1
\end{bmatrix} \quad B = \begin{bmatrix}
.0990 \\
.005 \\
.0497
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix} \quad D = \begin{bmatrix}
0
\end{bmatrix}
\]

• Now, write the objective function:

\[
J = \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U - 2 \Delta U^T \Phi^T (R_s - F x(k_i))
\]

where,

\[
\Phi^T \Phi = \begin{bmatrix}
.1760 & .1553 & .1361 \\
.1553 & .1373 & .1204 \\
.1361 & .1204 & .1057
\end{bmatrix}; \quad \Phi^T F = \begin{bmatrix}
.1972 & -.1758 & 1.4187 \\
.1740 & -.1552 & 1.2220 \\
.1522 & -.1359 & 1.0443
\end{bmatrix}
\]

and

\[
R_s = \begin{bmatrix}
1.4187 \\
1.2220 \\
1.0443
\end{bmatrix} \times r(k_i)
\]

• Select observer poles at \( \{0 \; 0 \; 0\} \)

• Computing the closed-loop compensated eigenvalues for a range of control weighting values \( (r_w) \) results in the following values for \( K_{mpc}(r_w) \):

\[
K_{mpc}(.01) = \begin{bmatrix}
13.1552 & 84.1009 & 3.7417
\end{bmatrix}
\]

\[
K_{mpc}(.1) = \begin{bmatrix}
11.3704 & 58.4467 & 1.1666
\end{bmatrix}
\]

\[
K_{mpc}(1.0) = \begin{bmatrix}
9.5534 & 43.5058 & 0.6567
\end{bmatrix}
\]
\[ K_{mpc}(10) = \begin{bmatrix} 5.5692 & 14.6499 & 0.2460 \\ \end{bmatrix} \]
\[ K_{mpc}(100) = \begin{bmatrix} 3.6823 & 3.0537 & 0.0868 \\ \end{bmatrix} \]

- These give rise to the corresponding closed-loop eigenvalues:

\[ \lambda (.01) = \begin{Bmatrix} .3357 + j0.4 & .3357 - j0.4 & 0.3824 \end{Bmatrix} \]
\[ \lambda (.1) = \begin{Bmatrix} .3654 + j0.2650 & .3654 - j0.2650 & 0.7552 \end{Bmatrix} \]
\[ \lambda (1.0) = \begin{Bmatrix} .4697 + j0.3063 & .4697 - j0.3063 & 0.8262 \end{Bmatrix} \]
\[ \lambda (10) = \begin{Bmatrix} .7710 + j0.2439 & .7710 - j0.2439 & 0.7819 \end{Bmatrix} \]
\[ \lambda (100) = \begin{Bmatrix} .9240 + j0.1610 & .9249 - j0.1610 & .7283 \end{Bmatrix} \]
• The output response to a unit set-point change is depicted in the following figure for each of the control weighting values.

• The corresponding control inputs are shown in the plot below.
• Let us now establish our problem constraint as:

\[
\begin{bmatrix}
1 & 0 & 0 \\
-1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\Delta u(k_i) \\
\Delta u(k_i + 1) \\
\Delta u(k_i + 2)
\end{bmatrix} \leq
\begin{bmatrix}
3.0 \\
-1.5
\end{bmatrix}
\]

\[M \Delta U \leq \gamma\]

• Recall that we constructed our original cost function as:

\[J = (R_s - Y)^T(R_s - Y) + \Delta U^T \bar{R} \Delta U\]

where

\[Y = Fx(k_i) + \Phi \Delta U\]

• Substituting and multiplying out we obtained:

\[J = (R_s - Fx(k_i))^T(R_s - Fx(k_i)) - 2\Delta U^T \Phi^T(R_s - Fx(k_i)) + \Delta U^T(\Phi^T \Phi + \bar{R}) \Delta U\]

• We found the optimizing control input sequence from the stationarity condition \(\frac{\partial J}{\partial \Delta U} = 0\) as:

\[\Delta U^* = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T(R_s - Fx(k_i))\]

• Using this expression with the assumption that the initial state \(x(0) = 0\), we obtain

\[\Delta U = \begin{bmatrix} 3.7417 & -6.2794 & 2.7112 \end{bmatrix}^T\]

Here we see that the constraint is violated since

\[\Delta u(1) = 3.7417 \neq 3.0\]

• Using Hildreth’s quadratic programming algorithm we obtain

\[\Delta U = \begin{bmatrix} 3.000 & -4.7241 & 1.8843 \end{bmatrix}\]
Using the first component, $\Delta u(1) = 3.000$ and $y(1) = 0$, the new estimated state variable is

$$x(2) = \begin{bmatrix} .2791 \\ .0149 \\ .1491 \end{bmatrix}$$

Computing the next incremental input sequence gives

$$\Delta U^* = \begin{bmatrix} -1.9777 & -4.1326 & 3.3456 \end{bmatrix}$$

and the constrained solution via Hildreth's algorithm gives

$$\Delta U^*_c = \begin{bmatrix} -1.5 & -5.1342 & 3.8781 \end{bmatrix}$$

This generates the new state update

$$x(3) = \begin{bmatrix} .1367 \\ .0366 \\ .5155 \end{bmatrix}$$

Continuing,

$$\Delta U^* = \begin{bmatrix} -3.0671 & -0.6187 & 2.4816 \end{bmatrix}$$

$$\Delta U^*_c = \begin{bmatrix} -1.5 & -3.9045 & 4.2285 \end{bmatrix}$$

and

$$x(4) = \begin{bmatrix} -.0261 \\ .0422 \\ .9372 \end{bmatrix}$$

And again

$$\Delta U^* = -\begin{bmatrix} -2.9689 & 2.2251 & 1.0865 \end{bmatrix}$$

$$\Delta U^*_c = \begin{bmatrix} -1.5 & -.8547 & 2.7239 \end{bmatrix}$$
and the updated state,

\[
x(5) = \begin{bmatrix}
-0.1865 \\
0.0315 \\
1.2523
\end{bmatrix}
\]

- So we obtain the optimal incremental input sequence for the first four time samples as:

\[
\Delta U^* = \begin{bmatrix}
3.0 \\
-1.5 \\
-1.5 \\
-1.5 \\
\end{bmatrix}
\]

giving the output sequence:

\[
Y = \begin{bmatrix}
0 \\
0.1491 \\
0.5155 \\
0.9372 \\
1.2523
\end{bmatrix}
\]
(mostly blank)