Model Predictive Control System Design and Implementation Using MATLAB

Liuping Wang   Alexander Brown

Discrete-Time MPC for Beginners

Day-to-day Application Example of Predictive Control

The general design objective of model predictive control is to compute a trajectory of a future manipulated variable u to optimize the future behavior of the plant output y. The optimization is done within a limited time window by givign plant information at the start time of the window.

Principle of MPC

1. Moving horizon window
2. Prediction horizon
3. Receding horizon control
4. We require information at time t_i (know the state $x(t_i)$) in order to predict the future.
5. A given model that describes the dynamics of the system
6. A criterion that reflects the current objective

State Space Model with Embedded Integrator

SISO System

Assume a single-input and single-output system of the form:

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) \\ y(k) &= C_m x_m(k) \end{aligned}$$

By Taking the difference of $x_m(k) = A_m x_m(k-1) + B_m u(k-1)$ we get:

$$\begin{aligned} &\Delta x_m(k+1) = x_m(k+1) - x_m(k) \\ &\Delta u(k) = u(k) - u(k-1) \\ &\implies \Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) \end{aligned}$$

The next step is now to connect the $\Delta x_m(k)$ with the output $y(k)$. We do this by creating a new state variable $x(k) = [\Delta x_m(k)^T y(k)]^T$ giving:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k)$$

$$y(k) = \begin{bmatrix} o_m \\ 1 \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

Where $o_m = [00 \cdots 0]$. From this, we can see that we get an augmented model of the form $\dot{x} = Ax + Bu$.

Predictive Control within One Optimization Window

Now that we have a way of modeling the system, the next step in the design of a predictive control system is to calculate the predicted plant output with the future control signal. This prediction is described within an optimization window (receding horizon). Assume the current time is k_i, the length of the optimization window is N_p samples, and N_c is the control horizon dictating the number of parameters used to capture the future control trajectory. N_c is chosen to be less than or equal to N_p.

Prediction of State and Output Variables

The case of using observers is considered later, for now we assume we can view the state.

Based on the model given above, assume we "sensed" the state at time k_i. Let's begin by plugging that state to find $x(k+1)$:

$$y(k_i+1|k_i) = Ax(k_i) + CB\Delta(k_i)$$

Where the notation $y(k_i+1|k_i)$ indicates the estimated future state given the data sensed at k_i. Continuing this N_p steps and by plugging the previous estimated state as the next variable to propagate into the future another step (i.e. use the solution of $y(k_i+1|k_i)$ to find $y(k_i+2|k_i)$):

$$\begin{aligned}
&y(k_i+1|k_i) = CAx(k_i) + CB\Delta(k_i)\\
&y(k_i+2|k_i) = CA^2x(k_i) + CAB\Delta(k_i) + CB\Delta(k_i+1)\\
&y(k_i+3|k_i) = CA^3x(k_i) + CA^2B\Delta(k_i) + CAB\Delta(k_i+1) + CB\Delta(k_i+2)\\
&\vdots\\
&y(k_i+N_p|k_i) = CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta(k_i) + \cdots + CA^{N_p-N_c}B\Delta(k_i) + CB\Delta(k_i+N_c-1)
\end{aligned}$$

Rewriting this in a more recognizable and simple form:

$$Y = Fx(k_i) + \Phi\Delta U$$

Where Y is our solution matrix, ΔU is the matrix $[\Delta u(k_i)\ \Delta u(k_i+1)\ \cdots]$, F is the matrix $[CA\ CA^2\ CA^3\ \cdots]^T$,

Optimization

Much like with LQR, we define a cost function J that reflects the control object:

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

- The first term is linked to the objective of minimizing the error between the predicted output and the set-point signal.
- The second term is reflects the consideration given to the size of ΔU when the objective function J is made to be as small as possible
- $\bar{R}$ is a diagonal matrix in the form $r_w I_{N_c\times N_c}$ where r_w is used as a tuning parameter

Intuitively, the last bullet point above can be thought of this way: if $r_w = 0$, then we don't care how large ΔU is. We are only concerned at minimizing the first term as much as possible.

Expanding J and solving the optimization problem $\frac{\partial J}{\partial \Delta U} = 0$ we find:

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(\bar{R}_s r(k_i) - Fx(k_i))$$

where $r(k_i)$ is the set point signalA

Receding Horizon Control

In receding horizon control, although we calculate ΔU that contains the control for N_c steps, we only use the first sample of this sequence. When the next sample arrives, we repeat the process using the new state date that has been sensed. Repeating this in real time gives us the **receding control law**.

Closed-Loop Control System

Looking back at our optimal solution of ΔU

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

- $(\Phi^T\Phi + \bar{R})^{-1}(R_s)$ corresponds to the **set-point change**
- $-(\Phi^T\Phi + \bar{R})^{-1}(Fx(k_i))$ corresponds to the **state feedback control**

These matrices are constants for time-invariant systems. Because of the receding control law discussed above we only take the first element of ΔU

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc}x(k_i)$$

where $K_y = (\Phi^T\Phi + \bar{R})^{-1}(R_s)$ and $K_{mpc} = -(\Phi^T\Phi + \bar{R})^{-1}(Fx(k_i)$. Therefore:

$$x(k+1) = Ax(k) - BK_{mpc}x(k) + BK_y r(k) = (A - BK_{mpc})x(k) + BK_y r(k)$$

And the closed loop characteristic equation is found by $det[\lambda I - (A - BK_{mpc}] = 0$.

Note

> As a consequence of this structure, the last column of F is identical to $\bar{R}$, therefore K_y is identical to the last element of K_{mpc}.

Prediction Control of MIMO Systems

Consider the system

$$x_m(k+1) = A_m x_m(k) + B_m u(k) + B_d w(k)$$

$$y(k) = C_m x_m(k)$$

Where $w(k)$ is the input disturbance, assumed to be a sequence of integrated white noise. Much like before, taking the difference and augmenting the output state gives us:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{q\times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix} \epsilon(k)$$

$$y(k) = \begin{bmatrix} o_m & I_{q\times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}$$

- q is the number of outputs
- $\epsilon(k) = w(k) - w(k-1)$

Note

> Notice the diagonal nature of the new augmented matrix A. Therefore, the eigenvalues of A are the union of the eigenvalues of the diagonal.

Solution of Predictive Control for MIMO Systems

Taking the same steps as we did in the SISO derivation we can conclude the solution of minimizing J for ΔU being:

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}(\Phi^T \bar{R}_s r(k_i) - \Phi^T F x(k_i))$$

Applying the receding horizon control principle we find:

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc} x(k_i)$$

State Estimation

Just as we have seen in Linear Multi-variable Control:

$$\hat{x}(k+1) = A_m \hat{x}_m(k) + B_m u(k) + K_{ob}(y(k) - C_m \hat{x}_m(k))$$

State Estimate Predictive Control

Given the state estimation as shown above, the functional J is rewritten to be:

$$J = (R_s - F\hat{x}(k_i) + B\Delta u(k_i) + K_{ob}(y(k_i) - C\hat{x}(k_i))$$

For which we minimize over ΔU to find:

$$\Delta U = (\Phi^T\Phi + \bar{R})^{-1}\Phi^T(R_s - F\hat{x}(k_i))$$

which leads to

$$\Delta u(k_i) = K_y r(k_i) - K_{mpc}\hat{x}(k_i)$$

Combining the observer and augmented solution we find:

$$\begin{bmatrix} \tilde{(}k+1) \\ x(k+1) \end{bmatrix} = \begin{bmatrix} A - K_{ob}C & o_{n\times n} \\ -BK_{mpc} & A - BK_{mpc} \end{bmatrix} \begin{bmatrix} \tilde{x}(k) \\ x(k) \end{bmatrix} + \begin{bmatrix} o_{n\times m} \\ BK_y \end{bmatrix} r(k)$$

Discrete-Time MPC with Constrains

Formulation of Constrained Control Problem

The core idea is to modify Δu to suit the constraint that has been *activated.* This section discusses the operational constraints that are frequently uncounted in the design of control systems. These operational constraints are presented as linear inequalities of the control and plant variables.

Frequently Used Operational Constraints

Constraints on the Control Variable Incremental Variation: These are hard constraints on rate of change of the control variable.

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

Constraints on the Amplitude of the Control Variable: These are the most commonly encountered constraints.

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

Note

> $u(k)$ 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}. For example, if a valve is allowed to open in the range between 15% and 80% and its normal operating value is 30%, then $u^{min} = 15\% - 30\% = -15\%$ and $u^{max} = 80\% - 30\% = 50\%$.

Output Constraints: We can specify the operating range for the plant

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

Output variables are often implemented as "soft" constraints:

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

Output constraints often cause large changes in both the control and incremental variables when they are enforced (become active). When this happens, the control or incremental control variables can violate their constraints and the problem of constraint conflict occur.

Constraints as Part of the Optimal Solution

The key to translating these constraints into linear inequalities and relating them to the problem is to parameterize the constraint variables using the same parameter vector ΔU as the one used in the design of predictive control.

Traditionally the constraints are imposed for all future sampling instants. In the case of a manipulated variable constraint we write:

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

This can be re-written in a more compact matrix form:

$$\begin{gathered} -(C_1 u(k_i-1) + C_2\Delta U) \leq -U^{min} \\ (C_1 u(k_i-1) + C_2\Delta U) \leq U^{max} \end{gathered}$$

The output constraints are given as

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

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

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

subject to the inequality constraints

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

Where

$$M_1 = \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} -I \\ I \end{bmatrix}$$

$$M_3 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix}$$

$$N_1 = \begin{bmatrix} -U^{min} + C_1u(k_i - 1) \\ U^{max} - C_1u(k_i - 1) \end{bmatrix}$$

$$N_2 = \begin{bmatrix} -U^{min} \\ U^{max} \end{bmatrix}$$

$$N_3 = \begin{bmatrix} -Y^{min} + Fx(k_i) \\ Y^{max} - \ Fx(k_i) \end{bmatrix}$$

This is further compacted by writing

$$M\Delta U \leq \gamma$$

which is used to reference this hard constraint later on.

Note

$\Phi^T\Phi + \bar{R}$ is the Hessian matrix and is assumed to be positive definite.

Numerical Solutions to Quadratic Programming

Quadratic Programming for Equality Constraints

Consider the function

$$J = 1/2x^TEx + x^TF + \lambda^T(Mx - \gamma)$$

This form is known as the Lagrangian. This represents the same information as saying J subject to some set of constraints (as we did before). The constraint information can be extracted by taking the partial derivatives of the equation.

$$\frac{\partial J}{\partial x} = Ex + F + M^T\lambda = 0$$
$$\frac{\partial J}{\partial \lambda} = Mx - \gamma = 0$$

When represented in this form, it is easily observed that we are just solving a simple set of equations. Solving for λ and x gives us the optimal values:

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

Note

expanding $x = -E^{-1}F - E^{-1}M^T\lambda = x^0 - E^{-1}M^T\lambda$ shows us to terms. x^0 is the global optimal solution that gives a minimum of the original cost function J without constraints, the second term is a correction term due to the equality constraints.

Minimization with Inequality Constraints

In the minimization with inequality constraints, the number of constraints may be larger than the number of decision variables. Because we express these constraints with a $\leq$ we can have both *active* and *inactive* constraints (active when $M_i x = \gamma_i$ and inactive when $M_i x < \gamma_i$). The Kuhn-Tucker conditions define the active and inactive constraints in terms of Lagrange multipliers.

$$\begin{aligned} Ex + F + \textstyle\sum_{i \in S_{act}} \lambda_i M_i^T &= 0 \\ M_i x - \gamma_i &= 0 \quad i \in S_{act} \\ M_i x - \gamma_i &< 0 \quad i \notin S_{act} \\ \lambda &\geq 0 \quad i \in S_{act} \\ \lambda &= 0 \quad i \notin S_{act} \end{aligned}$$

Similarly to before, the optimal solution with inequality constraints has the closed-form

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

Active Set Method The essence of this method is to define a "working set". By this, reduce the constraint set to only the active constraints. The steps are as follows:

1. Solve equality constraint problem
2. If $\lambda_i \geq 0 \forall \lambda$ then the point is a local solution to the original problem
3. If $\lambda_i < 0$ then the objective function can be decreased by *relaxing* the constraint (in other words, delete the constraint from the equation because it is not active)

Primal-Dual Method

As of until now, we have been observing *primal* methods, this is where the solutions are based on the decision variables. In the active set methods, the active constraints need to be identified along with the optimal decision variables. If there are many constrains, the computational load can get quite high. The use of a *dual* method can be used to systematically identify the constraints that are not active. The dual problem is formulated as follows:

$$\underset{\lambda \geq 0}{max}\, \underset{x}{min}[1/2x^T Ex + x^T F + \lambda^T(Mx - \gamma)]$$

As we have seen before, the unconstrained solution is given to be $x = -E^{-1}(F + M^T\lambda)$. Substituting this into the equation above we get:

$$\underset{\lambda \geq 0}{max}(1/2\lambda^T H\lambda + \lambda^T K + 1/2\gamma^T E^{-1}\gamma)$$

Therefore:

$$J = 1/2\lambda^T H\lambda + \lambda^T K + 1/2\gamma^T E^{-1}\gamma$$

subject to $\lambda \geq 0$, denoted λ_{act}. The primal vector x is obtained by:

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

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

Hildreth's Quadratic Programming Procedure

In this algorithm, the direction vectors were selected to be the basis vector. The method is explicitly shown as:

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

with

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

Where ij represents the row/column respectively.

Closed-Form Solution of $\lambda*$

If we knew beforehand all of the active constraints, we could simply solve

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

to give us all of our optimal solutions.

Predictive Control with Constraints on Input Variables

- View book to read through examples (pg 69)

Discrete-Time MPC Using Laguerre Functions