ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

Simple Discrete-Time Model-Based Predictive Control (MPC)

Simple Discrete-Time MPC

- Discretized model of a plant,

$$\begin{aligned} \boldsymbol{x}[k+1] &= \boldsymbol{\Phi}\boldsymbol{x}[k] + \boldsymbol{\Gamma}\boldsymbol{u}[k] \\ \boldsymbol{y}[k] &= \boldsymbol{C}\boldsymbol{x}[k], \end{aligned}$$

  where $\boldsymbol{\Phi} \in \mathbb{R}^{n\times n}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{n\times m}$, and $\boldsymbol{C} \in \mathbb{R}^{p\times n}$
- Apply the backward difference operator to the plant model

$$\Delta\boldsymbol{x}[k+1] = \boldsymbol{x}[k+1] - \boldsymbol{x}[k]$$

  to obtain

$$\Delta\boldsymbol{x}[k+1] = \boldsymbol{\Phi}\Delta\boldsymbol{x}[k] + \boldsymbol{\Gamma}\Delta\boldsymbol{u}[k]$$

  where $\Delta\boldsymbol{u}[k+1] = \boldsymbol{u}[k+1] - \boldsymbol{u}[k]$
- Backward difference to the plant's output

$$\begin{aligned} \Delta\boldsymbol{y}[k+1] &= \boldsymbol{y}[k+1] - \boldsymbol{y}[k] \\ &= \boldsymbol{C}\boldsymbol{x}[k+1] - \boldsymbol{C}\boldsymbol{x}[k] \\ &= \boldsymbol{C}\Delta\boldsymbol{x}[k+1] \end{aligned}$$

Augmented model of the plant

- Substituting

$$\Delta \boldsymbol{y}[k+1] = \boldsymbol{C}\Phi\Delta\boldsymbol{x}[k] + \boldsymbol{C}\Gamma\Delta\boldsymbol{u}[k]$$

- Hence,

$$\boldsymbol{y}[k+1] = \boldsymbol{y}[k] + \boldsymbol{C}\Phi\Delta\boldsymbol{x}[k] + \boldsymbol{C}\Gamma\Delta\boldsymbol{u}[k]$$

- Combining into one equation

$$\begin{bmatrix} \Delta\boldsymbol{x}[k+1] \\ \boldsymbol{y}[k+1] \end{bmatrix} = \begin{bmatrix} \Phi & \boldsymbol{O} \\ \boldsymbol{C}\Phi & \boldsymbol{I}_p \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x}[k] \\ \boldsymbol{y}[k] \end{bmatrix} + \begin{bmatrix} \Gamma \\ \boldsymbol{C}\Gamma \end{bmatrix} \Delta\boldsymbol{u}[k]$$

Model in a compact format

- Represent the plant output $\boldsymbol{y}[k]$ as

$$\boldsymbol{y}[k] = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_p \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x}[k] \\ \boldsymbol{y}[k] \end{bmatrix}$$

- Define the augmented state vector

$$\boldsymbol{x}_a[k] = \begin{bmatrix} \Delta\boldsymbol{x}[k] \\ \boldsymbol{y}[k] \end{bmatrix}$$

- Let

$$\boldsymbol{\Phi}_a = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{O} \\ \boldsymbol{C\Phi} & \boldsymbol{I}_p \end{bmatrix}, \quad \boldsymbol{\Gamma}_a = \begin{bmatrix} \boldsymbol{\Gamma} \\ \boldsymbol{C\Gamma} \end{bmatrix}, \quad \text{and } \boldsymbol{C}_a = \begin{bmatrix} \boldsymbol{O} & \boldsymbol{I}_p \end{bmatrix}$$

- Model in a compact format

$$\begin{aligned} \boldsymbol{x}_a[k+1] &= \boldsymbol{\Phi}_a\boldsymbol{x}_a[k] + \boldsymbol{\Gamma}_a\Delta\boldsymbol{u}[k] \\ \boldsymbol{y}[k] &= \boldsymbol{C}_a\boldsymbol{x}_a[k], \end{aligned}$$

where $\boldsymbol{\Phi}_a \in \mathbb{R}^{(n+p)\times(n+p)}$, $\boldsymbol{\Gamma}_a \in \mathbb{R}^{(n+p)\times m}$, and $\boldsymbol{C}_a \in \mathbb{R}^{p\times(n+p)}$

MPC: control action computed on-line

- Suppose the state vector $\boldsymbol{x}_a$ at each sampling time, k, is available to us
- Objective: construct a control sequence,

$$\Delta \boldsymbol{u}[k], \Delta \boldsymbol{u}[k+1], \ldots, \Delta \boldsymbol{u}[k+N_p-1],$$

  where N_p is the prediction horizon, such that a given cost function and constraints are satisfied
- The above control sequence will result in a predicted sequence of the state vectors,

$$\boldsymbol{x}_a[k+1|k], \boldsymbol{x}_a[k+2|k], \ldots, \boldsymbol{x}_a[k+N_p|k]$$

Computing control on-line

- Use predicted sequence of the state vectors,

$$\boldsymbol{x}_a[k+1|k], \boldsymbol{x}_a[k+2|k], \ldots, \boldsymbol{x}_a[k+N_p|k]$$

to compute predicted sequence of the plant's outputs,

$$\boldsymbol{y}[k+1|k], \boldsymbol{y}[k+2|k], \ldots, \boldsymbol{y}[k+N_p|k]$$

- Use the above information to compute the control sequence and then apply $\boldsymbol{u}[k]$ to the plant to generate $\boldsymbol{x}[k+1]$
- Repeat the process again, using $\boldsymbol{x}[k+1]$ as an initial condition to compute $\boldsymbol{u}[k+1]$, and so on
- Here $\boldsymbol{x}_a[k+r|k]$ denotes the predicted state at $k+r$ given $\boldsymbol{x}_a[k]$

Preparing to construct predicted control

- Constructing $\boldsymbol{u}[k]$ given $\boldsymbol{x}[k]$

$$
\begin{aligned}
\boldsymbol{x}_a[k+1|k] &= \boldsymbol{\Phi}_a \boldsymbol{x}_a[k] + \boldsymbol{\Gamma}_a \Delta \boldsymbol{u}[k] \\
\boldsymbol{x}_a[k+2|k] &= \boldsymbol{\Phi}_a \boldsymbol{x}_a[k+1|k] + \boldsymbol{\Gamma}_a \Delta \boldsymbol{u}[k+1] \\
&\quad = \boldsymbol{\Phi}_a^2 \boldsymbol{x}_a[k] + \boldsymbol{\Phi}_a \boldsymbol{\Gamma}_a \Delta \boldsymbol{u}[k] + \boldsymbol{\Gamma}_a \Delta \boldsymbol{u}[k+1] \\
&\vdots \\
\boldsymbol{x}_a[k+N_p|k] &= \boldsymbol{\Phi}_a^{N_p} \boldsymbol{x}_a[k] + \boldsymbol{\Phi}_a^{N_p-1} \boldsymbol{\Gamma}_a \Delta \boldsymbol{u}[k] + \cdots \\
&\quad + \boldsymbol{\Gamma}_a \Delta \boldsymbol{u}[k+N_p-1]
\end{aligned}
$$

Manipulations

- Represent the previous set of equations in the form,

$$
\begin{bmatrix} \boldsymbol{x}_a[k+1|k] \\ \boldsymbol{x}_a[k+2|k] \\ \vdots \\ \boldsymbol{x}_a[k+N_p|k] \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi}_a \\ \boldsymbol{\Phi}_a^2 \\ \vdots \\ \boldsymbol{\Phi}_a^{N_p} \end{bmatrix} \boldsymbol{x}_a[k]
$$

$$
+ \begin{bmatrix} \boldsymbol{\Gamma}_a & & & \\ \boldsymbol{\Phi}_a\boldsymbol{\Gamma}_a & \boldsymbol{\Gamma}_a & & \\ \vdots & & \ddots & \\ \boldsymbol{\Phi}_a^{N_p-1}\boldsymbol{\Gamma}_a & & \cdots & \boldsymbol{\Gamma}_a \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix}
$$

- Wish to design a controller that would force the plant output, $\boldsymbol{y}$, to track a given reference signal, $\boldsymbol{r}$

Compute the sequence of predicted outputs

$$
\begin{bmatrix} \boldsymbol{y}[k+1|k] \\ \boldsymbol{y}[k+2|k] \\ \vdots \\ \boldsymbol{y}[k+N_p|k] \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}_a \boldsymbol{x}_a[k+1|k] \\ \boldsymbol{C}_a \boldsymbol{x}_a[k+2|k] \\ \vdots \\ \boldsymbol{C}_a \boldsymbol{x}_a[k+N_p|k] \end{bmatrix}
$$

$$
= \begin{bmatrix} \boldsymbol{C}_a \boldsymbol{\Phi}_a \\ \boldsymbol{C}_a \boldsymbol{\Phi}_a^2 \\ \vdots \\ \boldsymbol{C}_a \boldsymbol{\Phi}_a^{N_p} \end{bmatrix} \boldsymbol{x}_a[k]
$$

$$
+ \begin{bmatrix} \boldsymbol{C}_a \boldsymbol{\Gamma}_a & & \\ \boldsymbol{C}_a \boldsymbol{\Phi}_a \boldsymbol{\Gamma}_a & \boldsymbol{C}_a \boldsymbol{\Gamma}_a & \\ \vdots & & \ddots & \\ \boldsymbol{C}_a \boldsymbol{\Phi}_a^{N_p-1} \boldsymbol{\Gamma}_a & & \cdots & \boldsymbol{C}_a \boldsymbol{\Gamma}_a \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}[k] \\ \Delta \boldsymbol{u}[k+1] \\ \vdots \\ \Delta \boldsymbol{u}[k+N_p-1] \end{bmatrix}
$$

Simplify the notation

Write the previous matrix equation compactly as

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{x}_a[k] + \boldsymbol{Z}\Delta\boldsymbol{U},$$

where

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}[k+1|k] \\ \boldsymbol{y}[k+2|k] \\ \vdots \\ \boldsymbol{y}[k+N_p|k] \end{bmatrix}, \quad \Delta\boldsymbol{U} = \begin{bmatrix} \Delta\boldsymbol{u}[k] \\ \Delta\boldsymbol{u}[k+1] \\ \vdots \\ \Delta\boldsymbol{u}[k+N_p-1] \end{bmatrix},$$

and

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{\Phi}_a \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^2 \\ \vdots \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^{N_p} \end{bmatrix}, \text{ and } \boldsymbol{Z} = \begin{bmatrix} \boldsymbol{C}_a\boldsymbol{\Gamma}_a & & & \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a\boldsymbol{\Gamma}_a & \boldsymbol{C}_a\boldsymbol{\Gamma}_a & & \\ \vdots & & \ddots & \\ \boldsymbol{C}_a\boldsymbol{\Phi}_a^{N_p-1}\boldsymbol{\Gamma}_a & & \cdots & \boldsymbol{C}_a\boldsymbol{\Gamma}_a \end{bmatrix}$$

The Performance Index

- Wish to construct a control sequence, $\Delta \boldsymbol{u}[k], \ldots, \Delta \boldsymbol{u}[k+N_p-1]$, that would minimize the cost

$$J(\Delta \boldsymbol{U}) = \frac{1}{2}\left(\boldsymbol{r}_p - \boldsymbol{Y}\right)^\top \boldsymbol{Q}\left(\boldsymbol{r}_p - \boldsymbol{Y}\right) + \frac{1}{2}\Delta \boldsymbol{U}^\top \boldsymbol{R} \Delta \boldsymbol{U},$$

  where $\boldsymbol{Q} = \boldsymbol{Q}^\top \succeq 0$ and $\boldsymbol{R} = \boldsymbol{R}^\top \succ 0$ are real symmetric positive semi-definite and positive-definite weight matrices, respectively
- The multiplying scalar, $1/2$, is just to make subsequent manipulations cleaner
- The vector $\boldsymbol{r}_p$ consists of the values of the command signal at sampling times, $k+1, k+2, \ldots, k+N_p$
- The selection of the weight matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$ reflects our control objective to keep the tracking error $\|\boldsymbol{r}_p - \boldsymbol{Y}\|$ "small" using the control actions that are "not too large"

Finding optimal control

- Apply the first-order necessary condition (FONC) test to $J(\Delta \boldsymbol{U})$,
$$\frac{\partial J}{\partial \Delta \boldsymbol{U}} = \mathbf{0}^{\top}.$$
- Solve the above equation for $\Delta \boldsymbol{U} = \Delta \boldsymbol{U}^{*}$, where
$$\begin{aligned}\frac{\partial J}{\partial \Delta \boldsymbol{U}} &= -\left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x}_a - \boldsymbol{Z}\Delta \boldsymbol{U}\right)^{\top} \boldsymbol{Q}\boldsymbol{Z} + \Delta \boldsymbol{U}^{\top} \boldsymbol{R} \\ &= \mathbf{0}^{\top}\end{aligned}$$
- Manipulate
$$-\boldsymbol{r}_p^{\top} \boldsymbol{Q}\boldsymbol{Z} + \boldsymbol{x}_a^{\top} \boldsymbol{W}^{\top} \boldsymbol{Q}\boldsymbol{Z} + \Delta \boldsymbol{U}^{\top} \boldsymbol{Z}^{\top} \boldsymbol{Q}\boldsymbol{Z} + \Delta \boldsymbol{U}^{\top} \boldsymbol{R} = \mathbf{0}^{\top}$$
- Transpose both sides of the above equation and rearranging terms
$$\left(\boldsymbol{R} + \boldsymbol{Z}^{\top} \boldsymbol{Q}\boldsymbol{Z}\right) \Delta \boldsymbol{U} = \boldsymbol{Z}^{\top} \boldsymbol{Q} \left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x}_a\right)$$

Checking for optimality

- The matrix $\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)$ is invertible, and in fact, positive definite because $\boldsymbol{R}=\boldsymbol{R}^{\top}\succ 0$ and $\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}$ is also symmetric and at least positive semi-definite
- Hence, $\Delta\boldsymbol{U}$ that satisfies the FONC is

$$\Delta\boldsymbol{U}^{*}=\left(\boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\right)^{-1}\boldsymbol{Z}^{\top}\boldsymbol{Q}\left(\boldsymbol{r}_{p}-\boldsymbol{W}\boldsymbol{x}_{a}\right)$$

- Apply the second derivative test to $J(\Delta\boldsymbol{U})$, which we refer to as the second-order sufficiency condition (SOSC)

$$\begin{aligned}\frac{\partial^{2}J}{\partial\Delta\boldsymbol{U}^{2}} &= \boldsymbol{R}+\boldsymbol{Z}^{\top}\boldsymbol{Q}\boldsymbol{Z}\\ &\succ 0,\end{aligned}$$

which implies that $\Delta\boldsymbol{U}^{*}$ is a strict minimizer of J.

MPC implementation