ECE 68000: MODERN AUTOMATIC CONTROL

Professor Stan Żak

MPC Design Using Discretized Model Directly

More comments on MPC

- MPC is also called the receding horizon control (RHC)
- MPC is a form of control where the current control action is obtained by solving on-line, at each sampling period, a finite horizon open-loop control problem, using the current state of the plant as the initial state
- The receding horizon corresponds to the usual behavior of the Earth's horizon: as one moves towards, it recedes, remaining a constant distance away from us*
- Nearly every application imposes constraints; actuators are naturally limited in the force they can apply, etc.
- MPC addresses constraints in a rigorous fashion

---

*J. M. Maciejewski, *Predictive Control With Constraints*, Prentice Hall, 2002

Computing the predicted control sequence

- Start with the discretized model of a given 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 $\boldsymbol{C} \in \mathbb{R}^{p\times n}$
- The state vector $\boldsymbol{x}$ at the sampling time, k, is available to us
- Objective: construct a control sequence,

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

  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,

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

Using the predicted sequence of the state vectors

- Use predicted sequence of the state vectors,

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

  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 $\boldsymbol{x}[k+r|k]$ denotes the predicted state at $k+r$ given $\boldsymbol{x}[k]$

Preparing to construct predicted sequence of control vectors

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

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

Represent equations in a matrix format

- Represent the previous set of equations in the form,

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

- 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

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

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

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

Simplify the notation

Write the previous matrix equation compactly as

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

where

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

and

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

The Performance Index

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

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

  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 $J(\boldsymbol{U})$,

$$\frac{\partial J}{\partial \boldsymbol{U}} = \mathbf{0}^\top.$$

- Solve the above equation for $\boldsymbol{U} = \boldsymbol{U}^*$, where

$$\begin{aligned}\frac{\partial J}{\partial \boldsymbol{U}} &= -\left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x}[k] - \boldsymbol{Z}\boldsymbol{U}\right)^\top \boldsymbol{Q}\boldsymbol{Z} + \boldsymbol{U}^\top \boldsymbol{R} \\ &= \mathbf{0}^\top\end{aligned}$$

- Manipulate

$$-\boldsymbol{r}_p^\top \boldsymbol{Q}\boldsymbol{Z} + \boldsymbol{x}[k]^\top \boldsymbol{W}^\top \boldsymbol{Q}\boldsymbol{Z} + \boldsymbol{U}^\top \boldsymbol{Z}^\top \boldsymbol{Q}\boldsymbol{Z} + \boldsymbol{U}^\top \boldsymbol{R} = \mathbf{0}^\top$$

- Transpose both sides of the above equation and rearranging terms

$$\left(\boldsymbol{R} + \boldsymbol{Z}^\top \boldsymbol{Q}\boldsymbol{Z}\right) \boldsymbol{U} = \boldsymbol{Z}^\top \boldsymbol{Q}\left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x}[k]\right)$$

Iterative first-order Lagrangian algorithm

- The first-order Lagrangian algorithm for the above optimization problem involving minimizing f subject to the inequality constraints, $\boldsymbol{g}(\boldsymbol{x}) \leq \mathbf{0}$,

$$\begin{aligned} \boldsymbol{x}^{[k+1]} &= \boldsymbol{x}^{[k]} - \alpha_k \left(\nabla f\left(\boldsymbol{x}^{[k]}\right) + D\boldsymbol{g}\left(\boldsymbol{x}^{[k]}\right)^{\top} \boldsymbol{\mu}^{[k]} \right) \\ \boldsymbol{\mu}^{[k+1]} &= \left[\boldsymbol{\mu}^{[k]} + \beta_k \boldsymbol{g}\left(\boldsymbol{x}^{[k]}\right)\right]_+ , \end{aligned}$$

  where the operation $[\cdot]_+ = \max(\cdot, 0)$ is applied component-wise

Using the Lagrangian algorithm in MPC implementation

- In our application to the MPC construction, the Lagrangian function is

$$l(\boldsymbol{U}, \boldsymbol{\mu}) = J\left(\boldsymbol{U}\right) + \boldsymbol{\mu}^\top \boldsymbol{g}\left(\boldsymbol{U}\right).$$

- The gradient of J with respect to $\boldsymbol{U}$

$$\nabla J\left(\boldsymbol{U}\right) = -\boldsymbol{Z}^\top \boldsymbol{Q}\left(\boldsymbol{r}_p - \boldsymbol{W}\boldsymbol{x} - \boldsymbol{Z}\boldsymbol{U}\right) + \boldsymbol{R}\boldsymbol{U}$$

- Suppose that we impose constraints on the plant output
- Then, the function $\boldsymbol{g}$ that represents these inequality constraints takes the form

$$\boldsymbol{g}\left(\boldsymbol{U}\right) = \begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{Z} \end{bmatrix} \boldsymbol{U} - \begin{bmatrix} -\boldsymbol{Y}^{\min} + \boldsymbol{W}\boldsymbol{x}[k] \\ \boldsymbol{Y}^{\max} - \boldsymbol{W}\boldsymbol{x}[k] \end{bmatrix}$$

Algorithm implementation

- The gradient of $\boldsymbol{\mu}^\top \boldsymbol{g}$ with respect to $\boldsymbol{U}$ is

$$\nabla\left(\boldsymbol{\mu}^\top \boldsymbol{g}\right) = \begin{bmatrix} -\boldsymbol{Z} \\ \boldsymbol{Z} \end{bmatrix}^\top \boldsymbol{\mu}.$$

- The first-order Lagrangian algorithm takes the form

$$\begin{aligned} \boldsymbol{U}^{(i+1)} &= \boldsymbol{U}^{(i)} - \alpha_i \left(\nabla J\left(\boldsymbol{U}^{(i)}\right) + \nabla\left(\boldsymbol{\mu}(i)^\top \boldsymbol{g}(i)\right)\right) \\ \boldsymbol{\mu}^{(i+1)} &= \left[\boldsymbol{\mu}^{(i)} + \beta_i \boldsymbol{g}\left(\boldsymbol{U}^{(i)}\right)\right]_+ \end{aligned}$$