ECE 68000: MODERN AUTOMATIC CONTROL

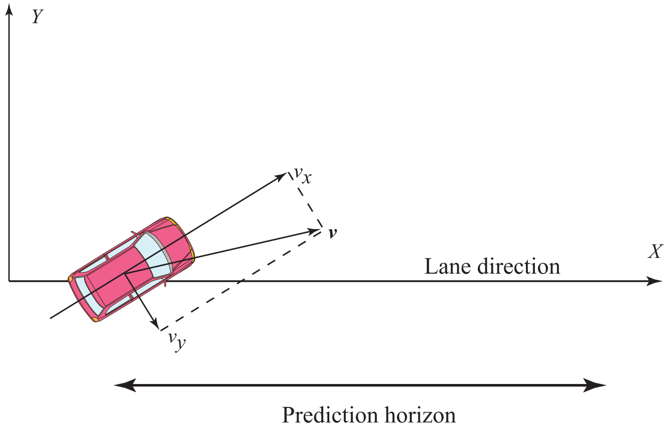
Professor Stan Żak

An Introduction to Model-Based Predictive Control (MPC)

Discrete Model-Based Predictive Controller (MPC)

- The model-based predictive control (MPC) methodology is also referred to as the moving horizon control or the receding horizon control
- The idea behind this approach can be explained using an example of driving a car
- The driver looks at the road ahead of him and taking into account the present state and the previous action predicts his action up to some distance ahead, which we refer to as the prediction horizon
- Based on the prediction, the driver adjusts the driving direction

The driver predicts future travel direction based on the current state of the car and the current position of the steering wheel



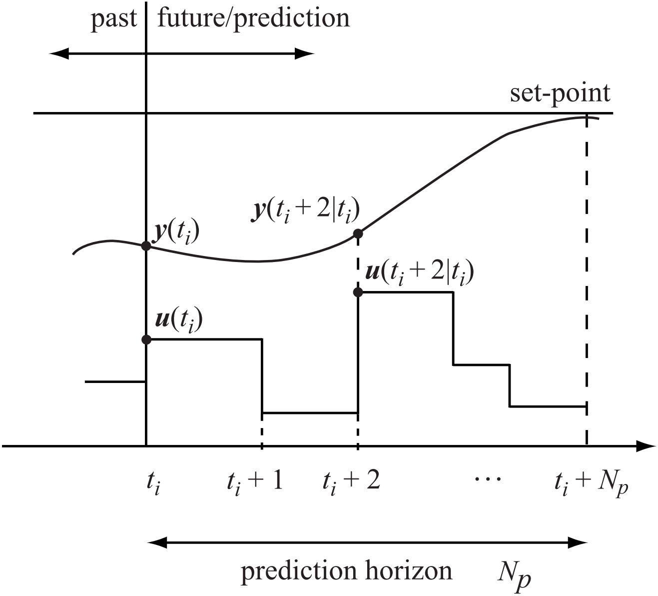
MPC approach: the current control action is computed on-line

- In the MPC approach, the current control action is computed on-line rather than using a pre-computed, off-line, control law
- A model predictive controller uses, at each sampling instant, the plant's current input and output measurements, the plant's current state, and the plant's model to
  - calculate, over a finite horizon, a future control sequence that optimizes a given performance index and satisfies constraints on the control action;
  - use the first control in the sequence as the plant's input

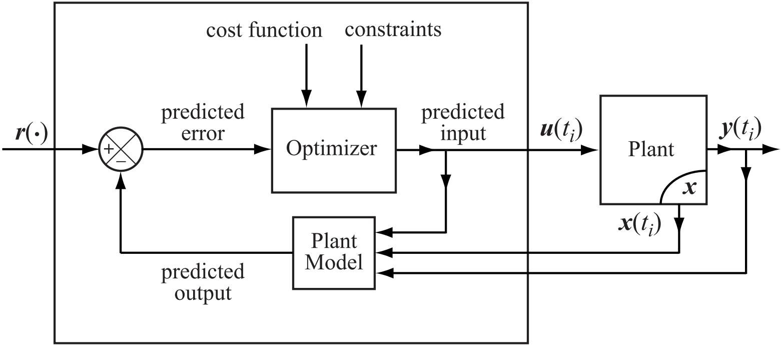
Lee and Markus' MPC idea in 1967

One technique for obtaining a feedback controller synthesis from knowledge of open-loop controllers is to measure the current control process state and then compute very rapidly for the open-loop control function. The first portion of this function is then used during the a short time interval, after which a new measurement of the process state is made and a new open-loop control function is computed for this new measurement. The procedure is then repeated. In this way external disturbances and other unknowns are taken into account in much the same way as is done by a feedback controller. If no disturbances or other unknowns are encountered, the recomputed control function should agree with the appropriate portion of the previously computed controller. This is essentially the principle of optimality [Bellman] in the theory of dynamic programming, a feedback principle.

Controller action construction using MPC



Basic Structure of MPC



Model Predictive Controller

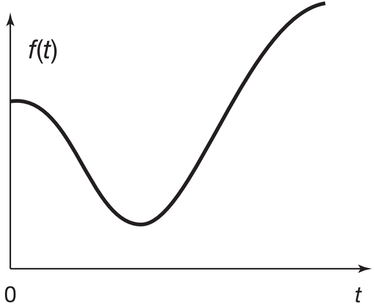
From a Continuous to a Discrete Plant Model

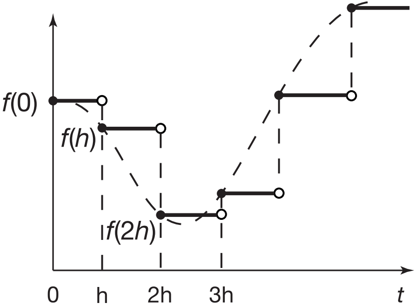
- Present a method for constructing linear discrete-time models from given linear continuous-time models
- The obtained discrete models will be used to perform computations to generate control commands
- Use a sample-and-hold device that transforms a continuous signal, $f(t)$, into the staircase signal, that is, piece-wise constant signal,

$$f(kh), \quad kh \leq t < [k+1]h,$$

  where h is the sampling period
- Use the sample and zero-order hold (ZOH) element

Sample and zero-order hold (ZOH) element operating on a continuous function





Exact discretization method development

- Given a continuous-time model,

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \quad \boldsymbol{x}_0 = \boldsymbol{x}(t_0) \\ \boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) \end{aligned}$$

- The solution to the state equation

$$\boldsymbol{x}(t) = e^{\boldsymbol{A}(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{\boldsymbol{A}(t-\tau)}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau.$$

- Assume that the input to the system is generated by a sample-and-hold device and has the form,

$$\boldsymbol{u}(t) = \boldsymbol{u}(kh), \quad kh \leq t < [k+1]h$$

- Let $t_0 = kh$ and $t = [k+1]h$ and let us use shorthand notation

$$\boldsymbol{x}(kh) = \boldsymbol{x}[k]$$

Manipulations

- Note that $\boldsymbol{u}(t) = \boldsymbol{u}(kh) \triangleq \boldsymbol{u}[k]$ is constant on the interval $[kh, [k+1]h)$, where the symbol, $\triangleq$, means "by definition"
- We have

$$\begin{aligned} \boldsymbol{x}[k+1] &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{kh}^{[k+1]h} e^{\boldsymbol{A}(kh+h-\tau)}\boldsymbol{B}\boldsymbol{u}[k]d\tau \\ &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{kh}^{[k+1]h} e^{\boldsymbol{A}(kh+h-\tau)}\boldsymbol{B}d\tau\, \boldsymbol{u}[k] \end{aligned}$$

- Consider the second term on the right-hand side of the above equation
- Let

$$\eta = kh + h - \tau.$$

Discretized model

- We have

$$
\begin{aligned}
\boldsymbol{x}[k+1] &= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{kh}^{[k+1]h} e^{\boldsymbol{A}(kh+h-\tau)}\boldsymbol{B}d\tau\, \boldsymbol{u}[k] \\
&= e^{\boldsymbol{A}h}\boldsymbol{x}[k] + \int_{0}^{h} e^{\boldsymbol{A}\eta}\boldsymbol{B}d\eta\, \boldsymbol{u}[k] \\
&= \boldsymbol{\Phi}\boldsymbol{x}[k] + \boldsymbol{\Gamma}\boldsymbol{u}[k],
\end{aligned}
$$

  where

$$
\boldsymbol{\Phi} = e^{\boldsymbol{A}h} \quad \text{and} \quad \boldsymbol{\Gamma} = \int_0^h e^{\boldsymbol{A}\eta}\boldsymbol{B}d\eta
$$

- The discrete-time (DT) output equation

$$
\boldsymbol{y}[k] = \boldsymbol{C}\boldsymbol{x}[k].
$$