NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Process Control: Part II- Model Predictive Control (EE6225, AY2019/20, S1)

Dr Xin Zhang

Assistant Professor

Office: NTU S2-B2c-95, Tel: 67905419

Lab: S2.2-B4-03

Email: Jackzhang@ntu.edu.sg

BASIC THEORY OF THE MODEL PREDICTIVE CONTROL (MPC): PART I

[17/10/2019]

- Main components of MPC
- Modelling of MPC
- MPC with state space model
- MPC with Carima model

BASIC THEORY OF THE MODEL PREDICTIVE CONTROL (MPC): PART I

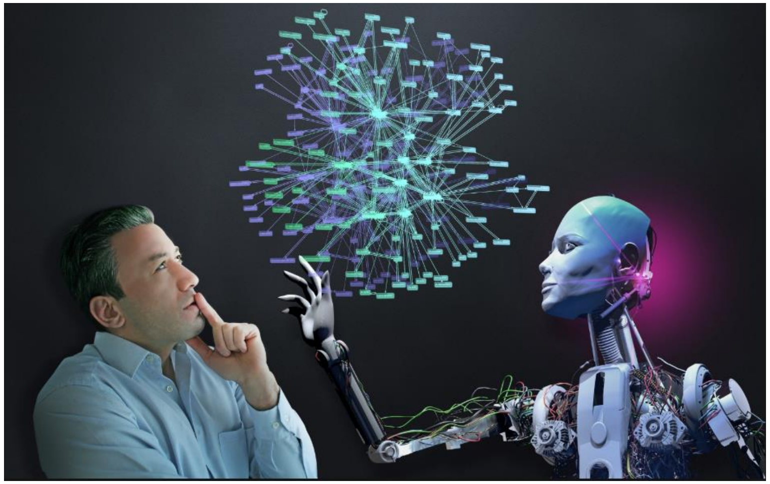
[17/10/2019]

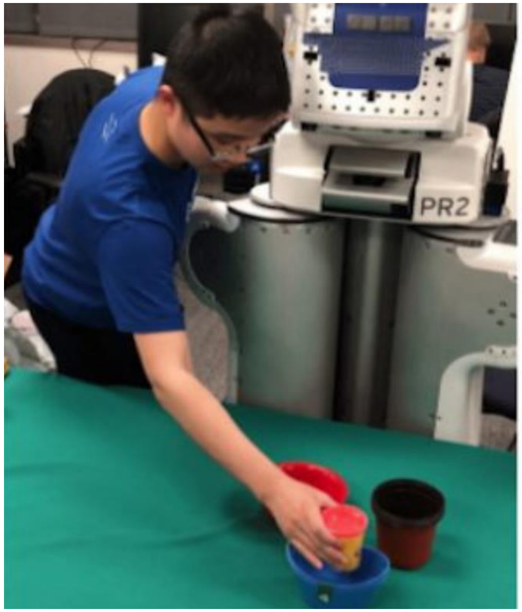
- Main components of MPC
- Modelling of MPC
- MPC with state space model
- MPC with Carima model

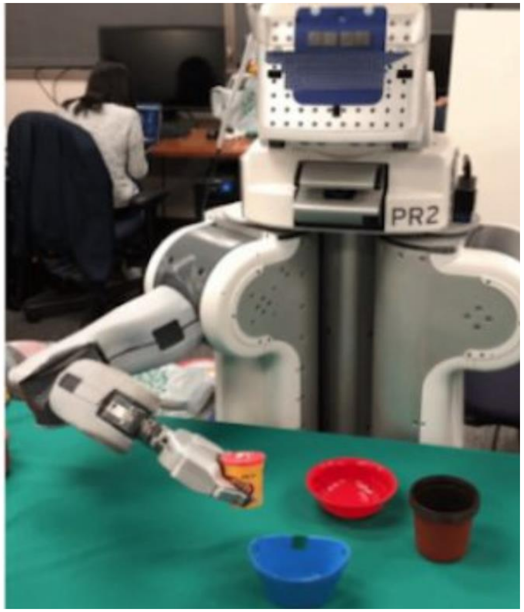
MPC mimics human's idea

- Using prediction and/or anticipation within control is logical-humans do this, naturally.

- Please consider how such concepts can be embedded into a MPC.







Main components of MPC

- Prediction
- Receding horizon
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

Main components of MPC

- Prediction
- Receding horizon
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

- Why prediction is important?
- How far should we predict?
- Consequences of not predicting?
- How do we predict?

WHY PREDICTION IS IMPORTANT?

We know prediction when we are kids

Parent will tell their children to think of possible consequences before act.

The results without prediction

Jumped of roof of shed
Broken ankle

Used knife incorrectly
Finger badly cut

Didn’t consult bus timetable -Missed bus

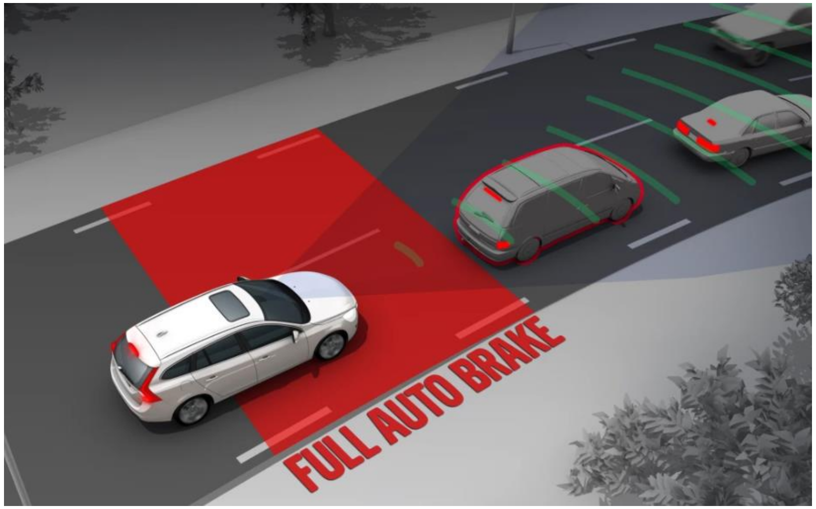
HOW FAR SHOULD WE PREDICT?

How far should we predict? (Drive)

- Prediction horizon is always asked in MPC,
- For human behavior, we all know how far for prediction.

1. Driving -- How far?

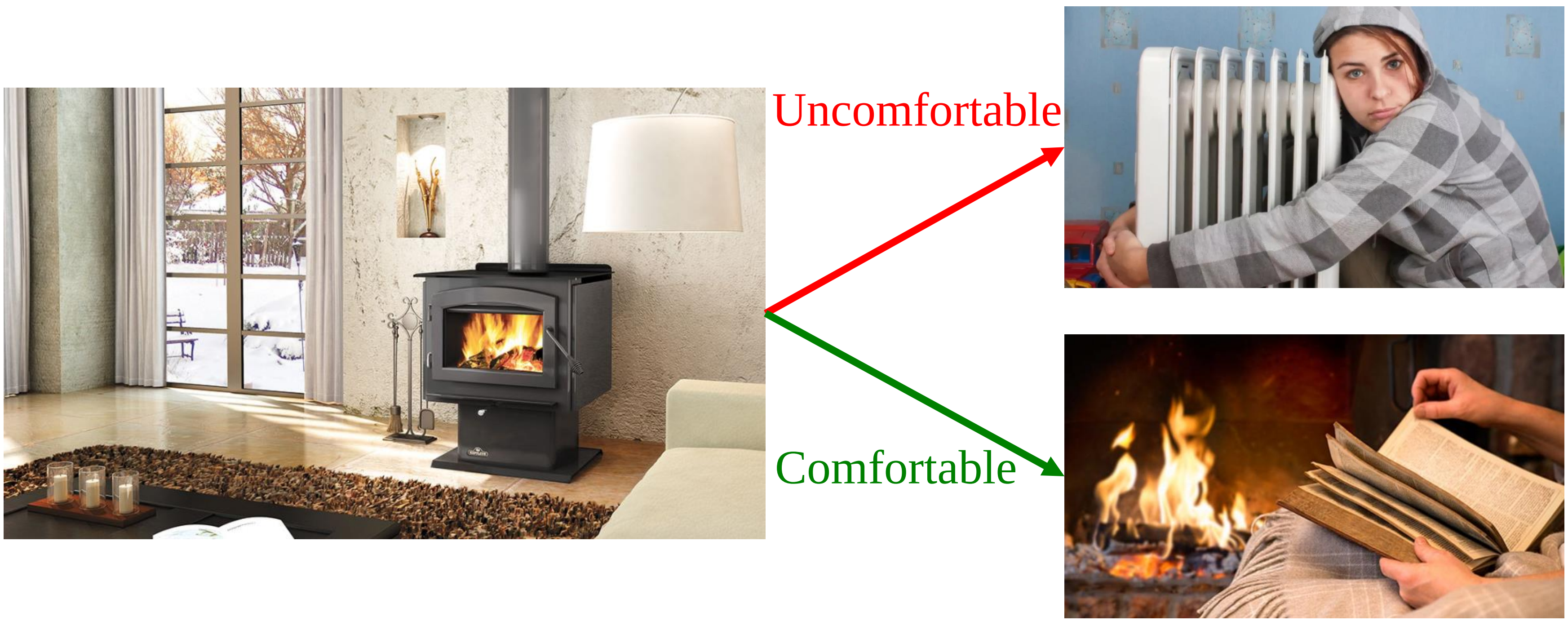
Beyond the safe braking distance, **or** a crash or accident happened.



How far should we predict? (Heating)

2. Heating a house -- How far?

Turn the heating on <u>far enough</u> in advance – <u>beyond</u> the <u>settling time</u>.

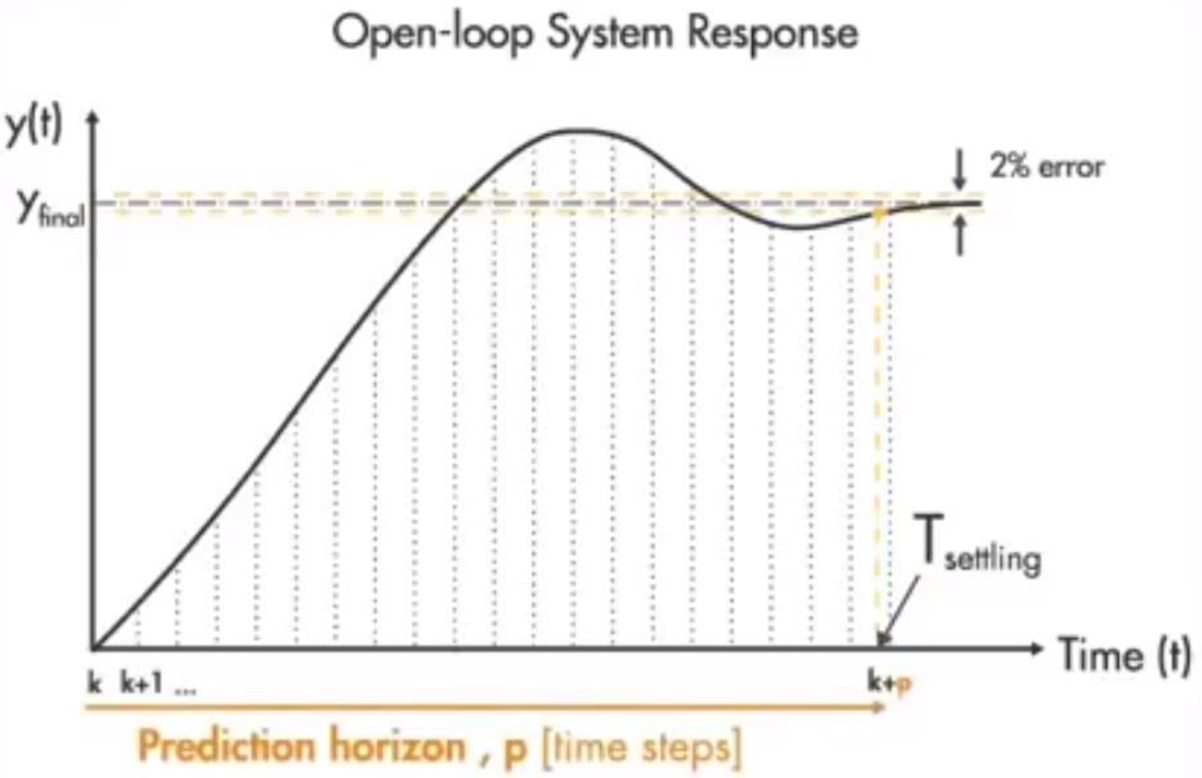


How far should we predict? (Moving item)

3. Moving heavy item -- How far?

Consider the whole trajectory, lifting, carrying and putting down again. **Or**, drop the item and cause damage.

The recommended predicting horizon



$T_{settling}$: Time it takes for the error $|y(t) - y_{final}|$ to fall to within 2% of y_{final}

$$\frac{T_r}{20} \leqslant T_s \leqslant \frac{T_r}{10}, \quad T_s\text{: Sample time}$$

$$p.T_s \geqslant T_{settling}$$

CONSEQUENCES OF NOT PREDICTING?

Consequences of not predicting

- We must predict beyond the key dynamics of a process;
- The missed issue could come back and bite us!

Driving

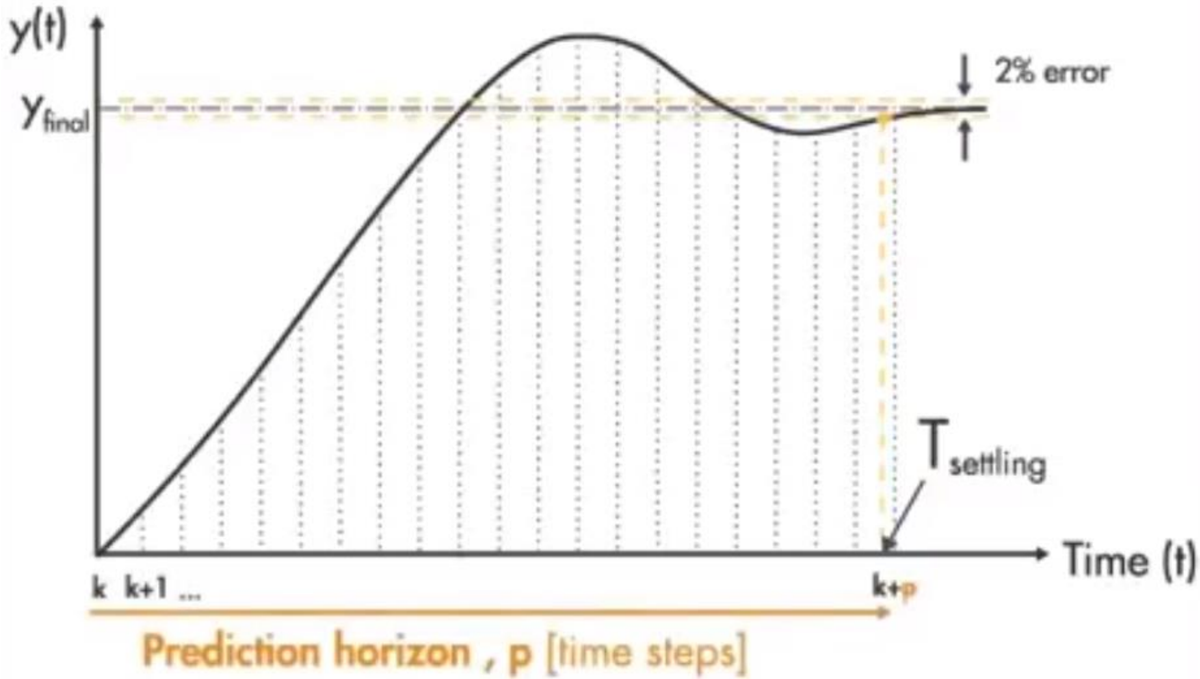
Heating

Moving item

HOW DO WE PREDICT?

Recall: the recommended predicting horizon

Open-loop System Response

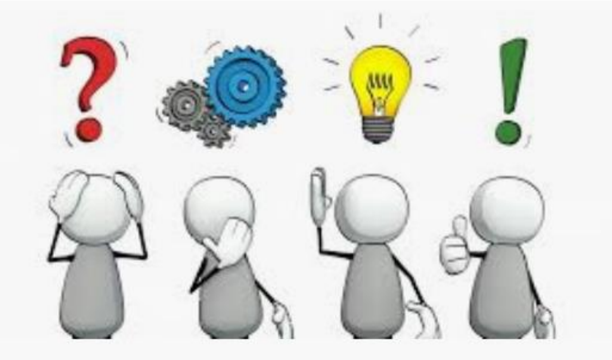


$T_{settling}$: Time it takes for the error $|y(t) - y_{final}|$ to fall to within 2% of y_{final}

$$\frac{T_r}{20} \leqslant T_s \leqslant \frac{T_r}{10}, \quad T_s\text{: Sample time}$$

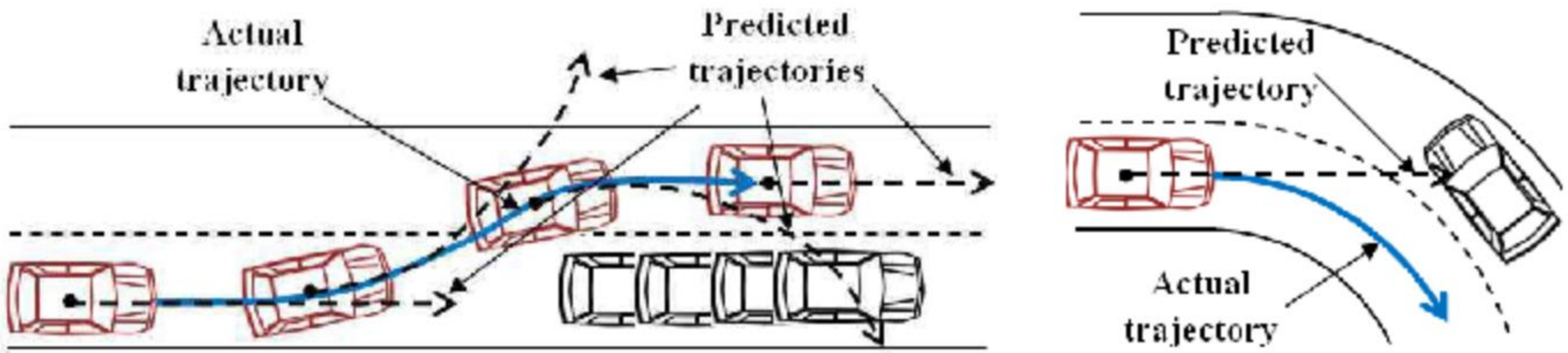
$$p.T_s \geqslant T_{settling}$$

- Prediction → make decision → prediction →….



- Prediction horizon > settling time;

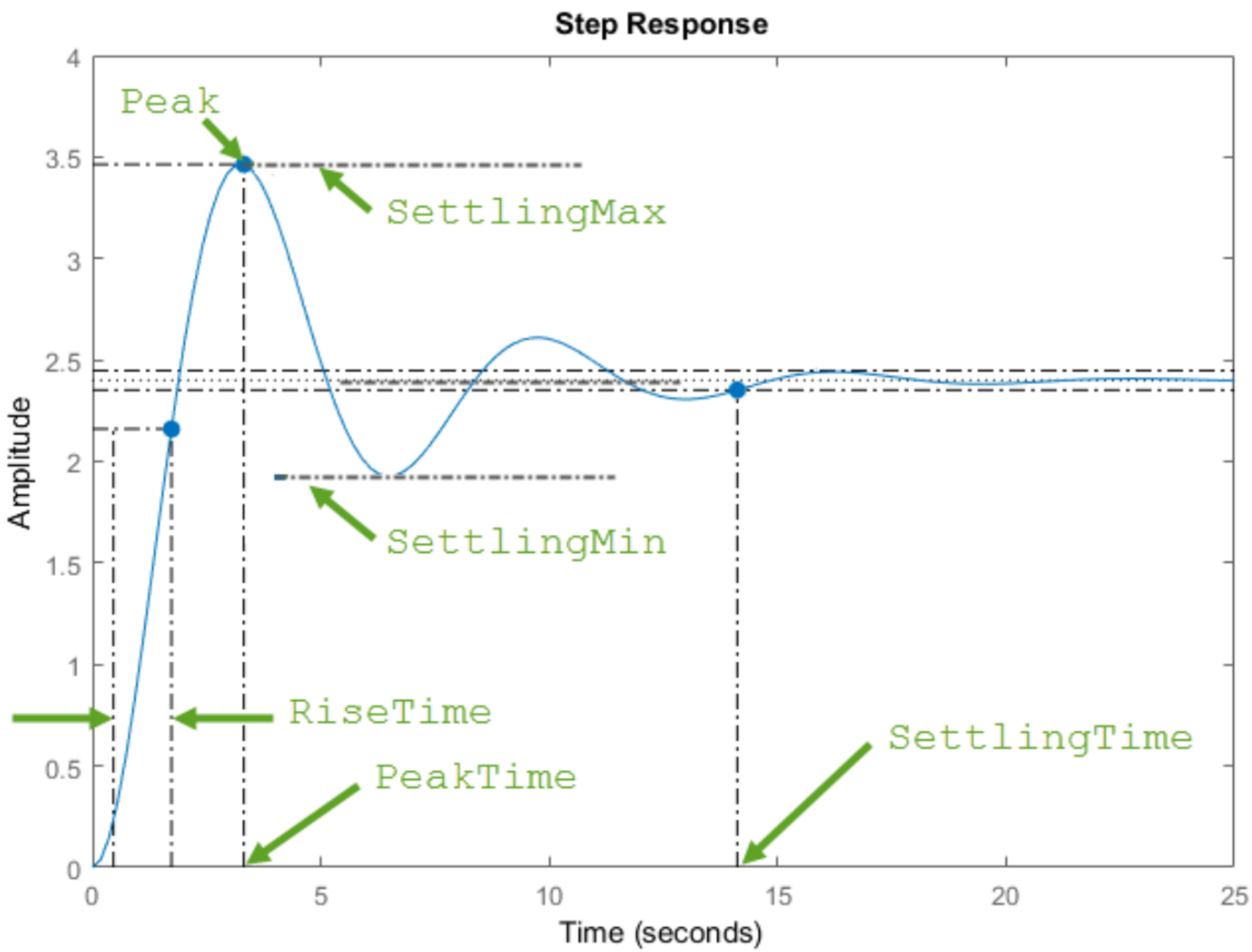
- The more accurate predictive the better.



(a) Lane changing (b) Entering a bend

What do we mean by accurate?

Evaluation index: steady-state error, fast transients, response, …



Main components of MPC

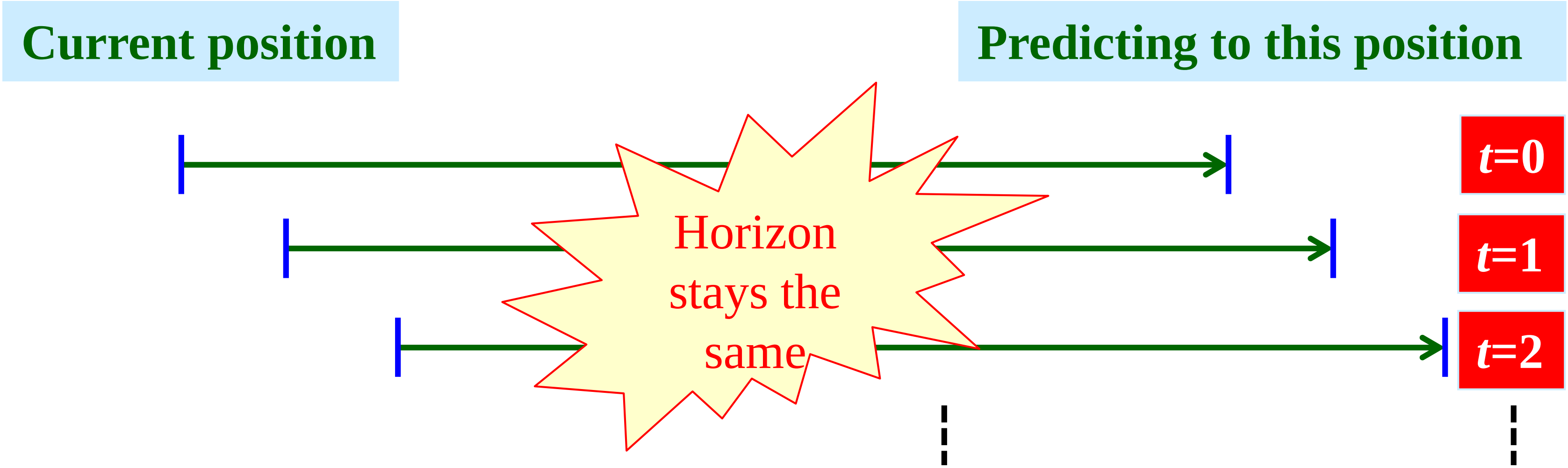
- Prediction
- **Receding horizon**
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

Receding horizon

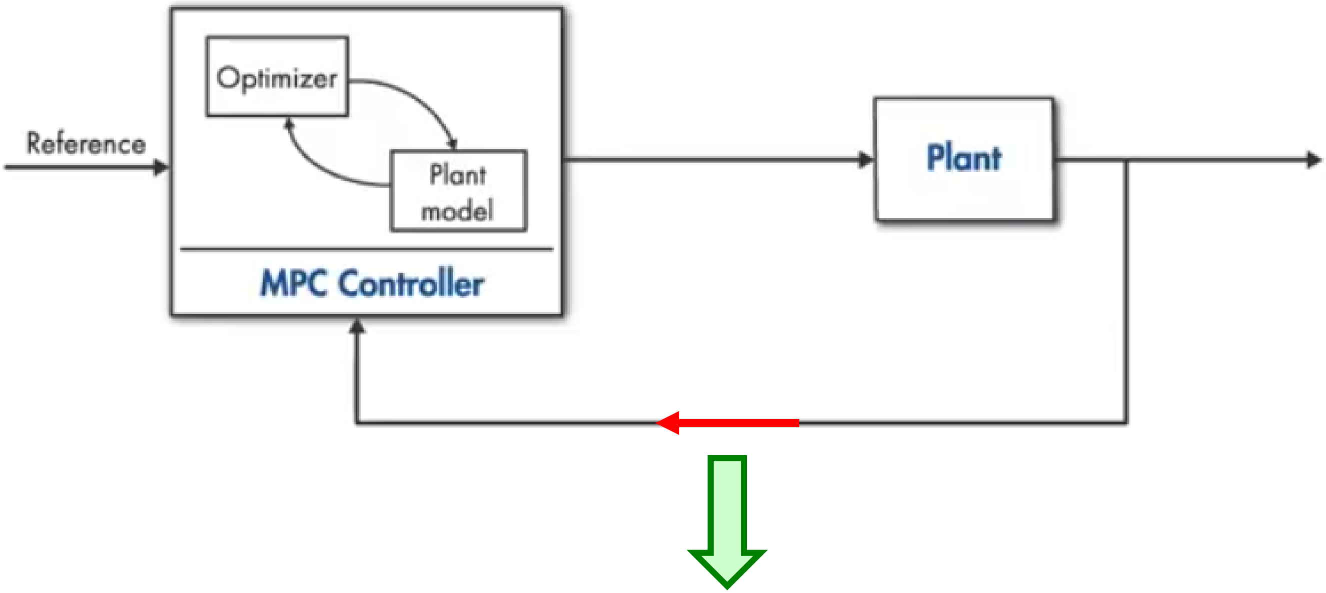
- Continually updating predictions and decision.
- Prediction horizon is relative to current position.



- The more accurate the better:

a) Measurement is a core part of a feedback loop.

b) Decisions via measurement is also important.



Feedback can guarantee the accuracy.

Main components of MPC

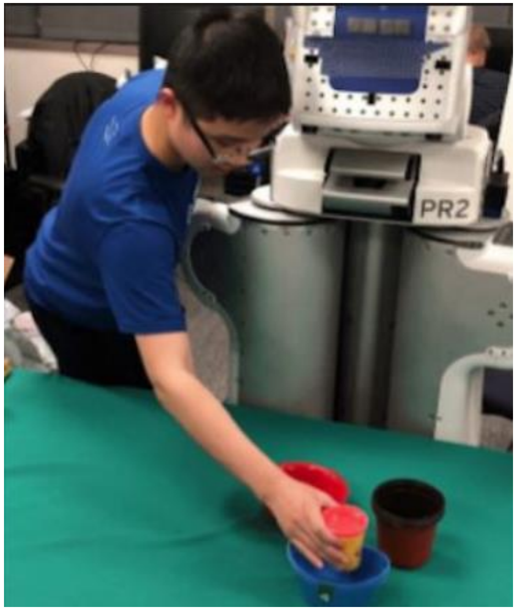
- Prediction
- Receding horizon
- **Modeling**
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

How MPC works?

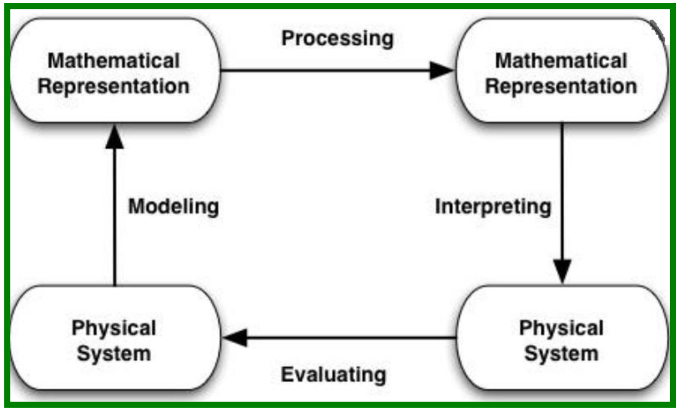
Should **not** attempt MPC design **before** we have the required understanding

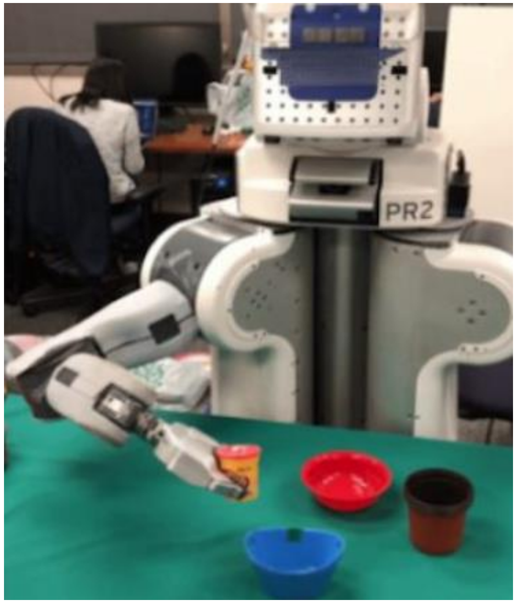
Modelling with MPC

- Motivation: modelling system (human) behavior.
- Task: How to define or determine an appropriate prediction model?



Modelling

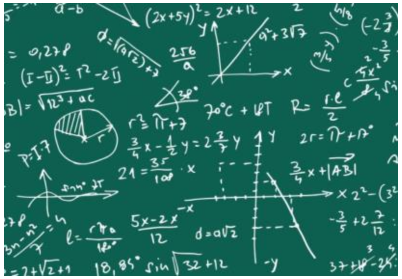




Modelling requirements

- Easy to use --- linear.

- Easy to identify parameters.



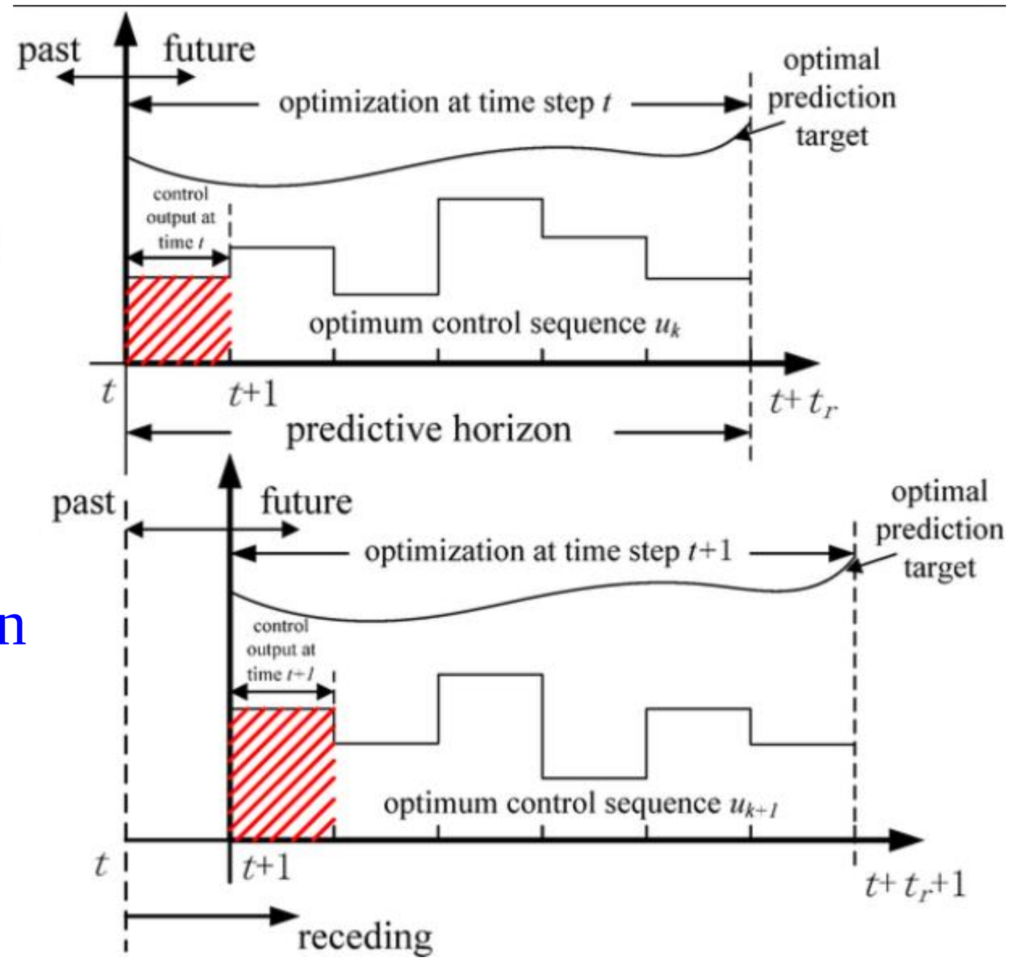
- Accurate predictions.

Perfect | Unacceptable | Acceptable | Unacceptable

One-step prediction is usually used

Most model are based on one-step ahead prediction errors!

Input → BLACK BOX → Output



- Black box
- One-step ahead prediction
- Simple and faster

Tips of Modelling

- The simplest model with accurate predictions is best.
- Practical accurate:
  a) 10%-20% error with the steady-state
  b) Can capture the key dynamic changes during transients.
- Rarely beneficial to improve accuracy with high order model.
- Feedback can correct small modelling errors.
- Long range prediction ability is required for MPC.

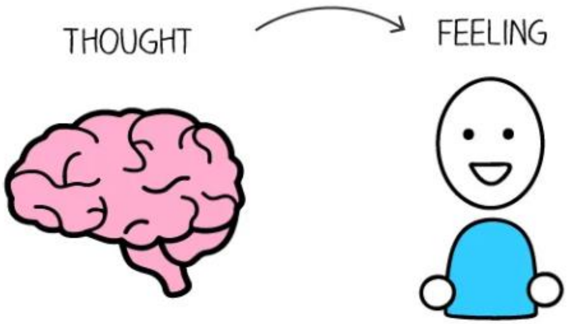
Main components of MPC

- Prediction
- Receding horizon
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

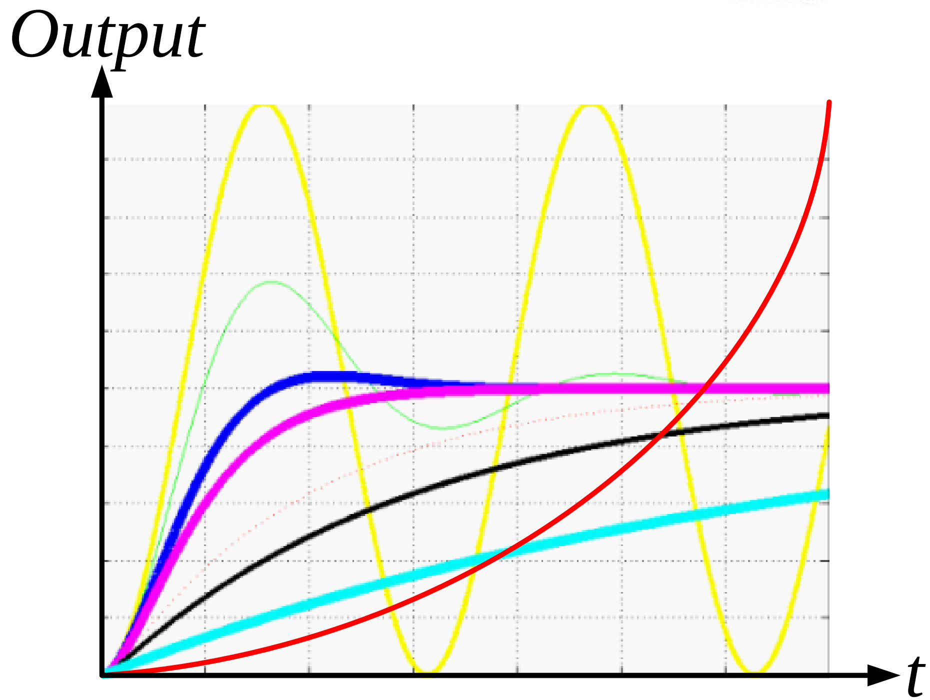
How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

Good or bad: rarely be quantitative



- Slow
- Oscillatory
- Unstable
- Ideal



The performance index concerns

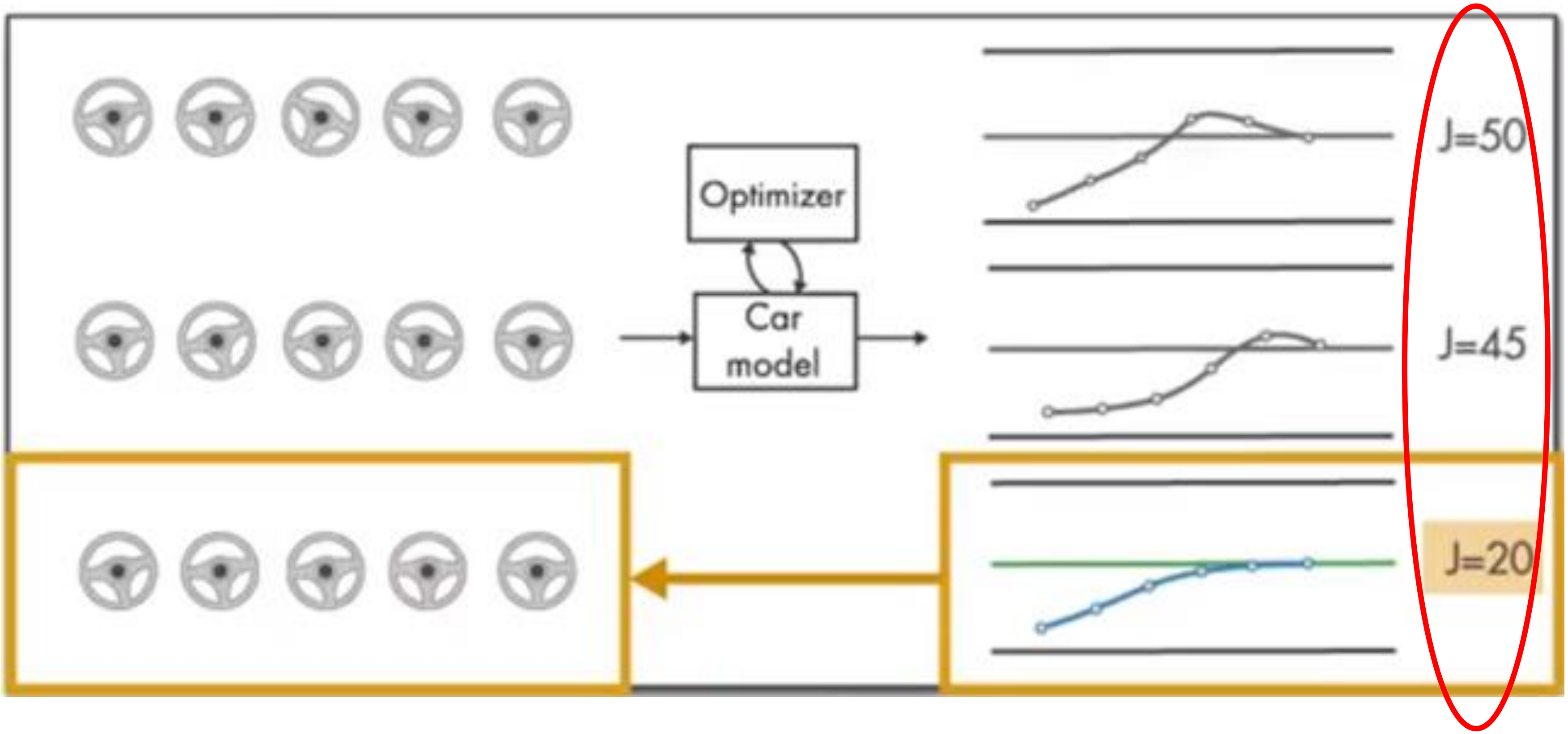
- What is the performance index used for?

- How should the performance index be designed?
- How can-do trade offs between optimal & safe/robust performance?

WHAT IS THE PERFORMANCE INDEX USED FOR?

How to use performance index?

- The performance index is a numeric definition for best.



HOW SHOULD THE PERFORMANCE INDEX BE DESIGNED?

How to design?

- Simpler definitions are better.
- Quadratic performance indices is preferred.

$$J = \frac{1}{2}\int_0 \left(x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru\right)\mathrm{d}t$$

EXAMPLE

$$\min_{u_k, x_k} \sum_k 0.5x_k^2 + 2u_k^2$$

HOW CAN-DO TRADE OFFS BETWEEN OPTIMAL & SAFE/ROBUST PERFORMANCE? (FROM HUMAN BEHAVIOR POINT OF VIEW)

- Beginner:
  - Simple strategy, - Get the ball back, - Anywhere & anyhow!

Tennis sports: Middle player

- Middle player
  - More complicated, - Get the ball back, - some methods of control and direction.

- Expert has:
  - Very complex; - Get the ball back; - Very precise on how (sometimes several shots ahead to create an opening)

Discussion on the tennis strategy selection

- Very complex strategy:
  - Get the ball back;
  - Very precise on how, i.e., hit the lines

Less Robust

- Simple strategy:
  - Only keeping the ball in play,
  - Aiming for the middle.

Who is more likely to make a mistake? [**Assume the opponent is passive**].

Same issues to driving (think of racing), cooking, robotics, etc.

Modelling and performance

Little experience **or** low-quality model

↓

<u>Cautious</u> performance index is realistic

Lots of experience **or** high-quality model

↓

<u>Ambitious</u> performance index is <u>possible</u>, but <u>no need</u>

High performance demand

- High performance demands are not cost free:

High performance implies high risk

↓

Less robustness for uncertainty

Low performance with means (low risk)

↓

Safe and robust to uncertainty

Main components of MPC

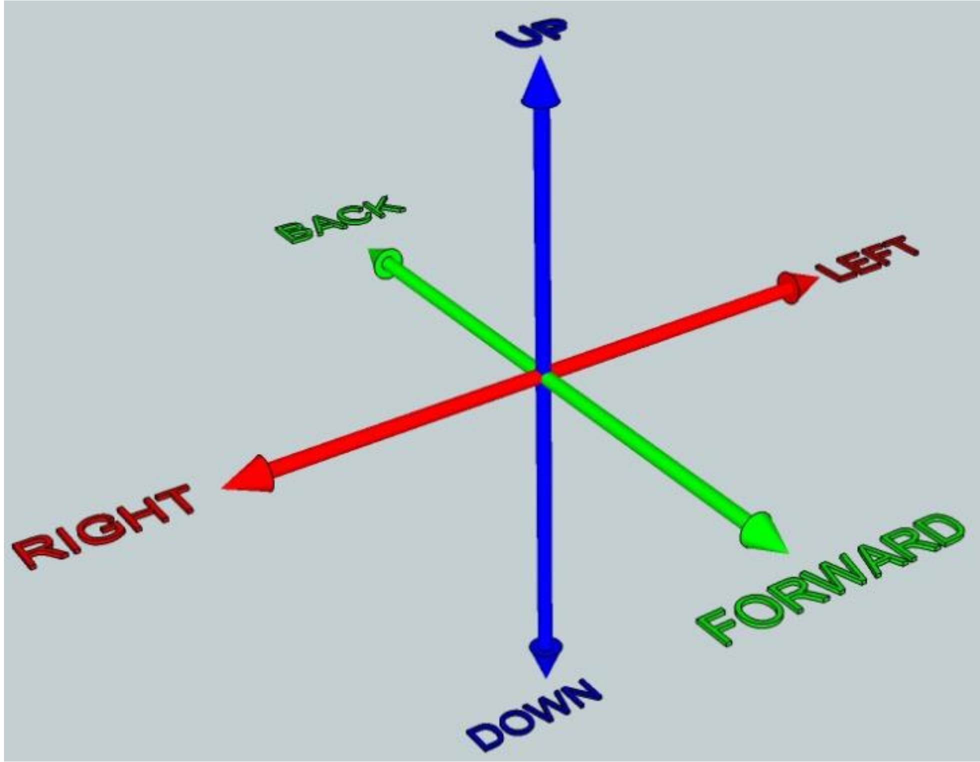
- Prediction
- Receding horizon
- Modeling
- Performance index
- **Degrees of freedom**
- Constraint handling
- Multivariable

How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

The concept of degrees of freedom (DOF)

- Degree of freedom (DOF) determines Prediction & control complexity

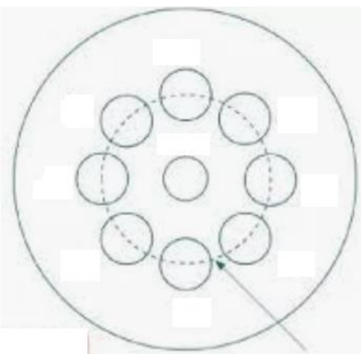


DOF = 3

- Bad model does not need <u>high performance index</u>
- Bad model does not need <u>high numbers of degrees</u>

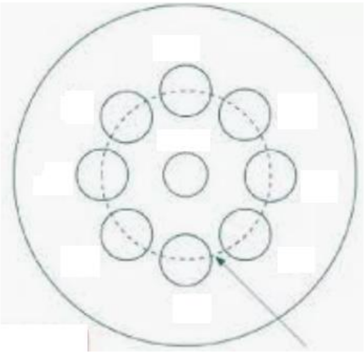
Example: Snooker

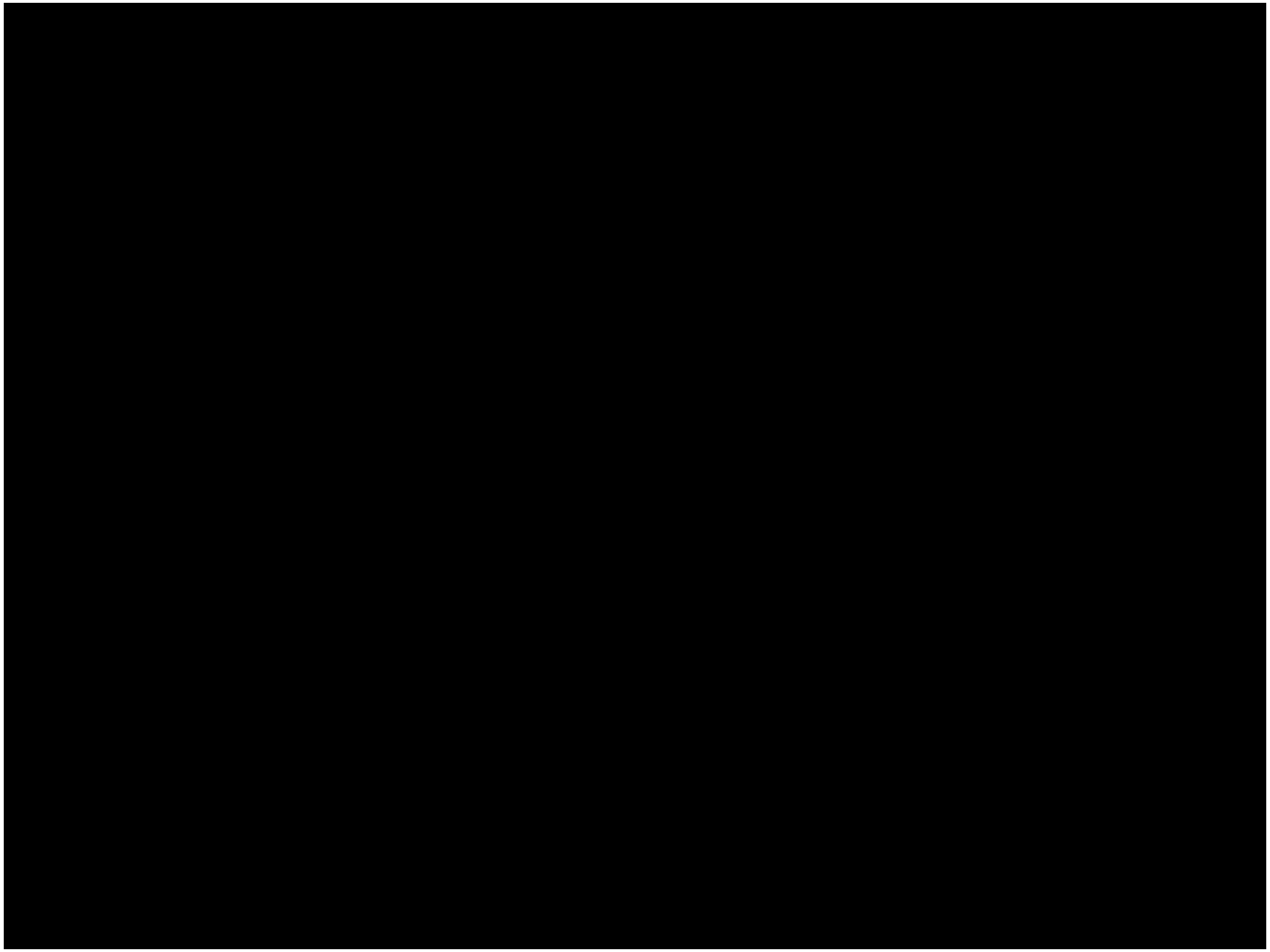
Hitting point



Cannot ask Snooker beginners play the master level!

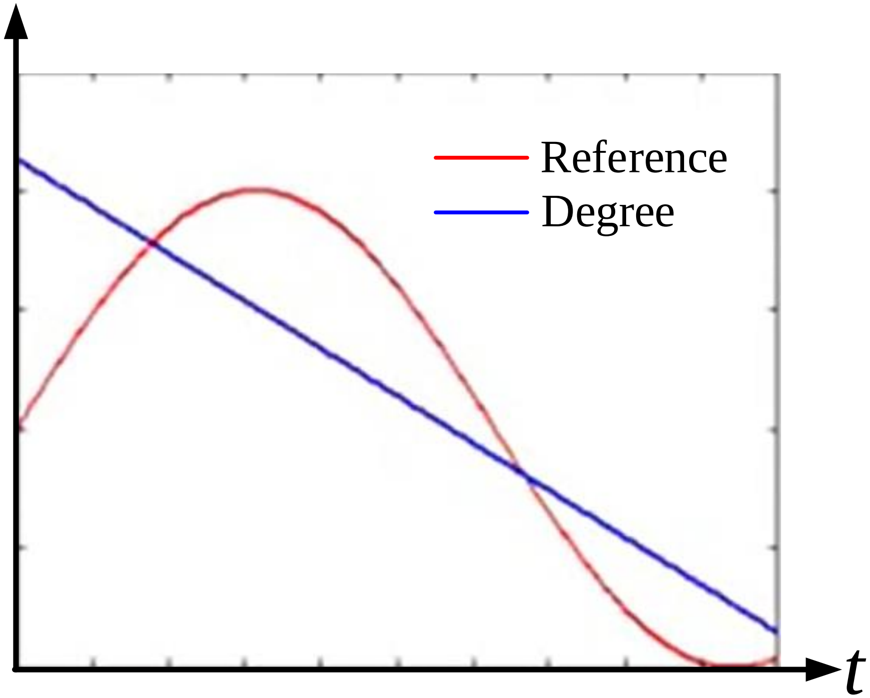
Example: Snooker





Simple DOF needs simple performance index

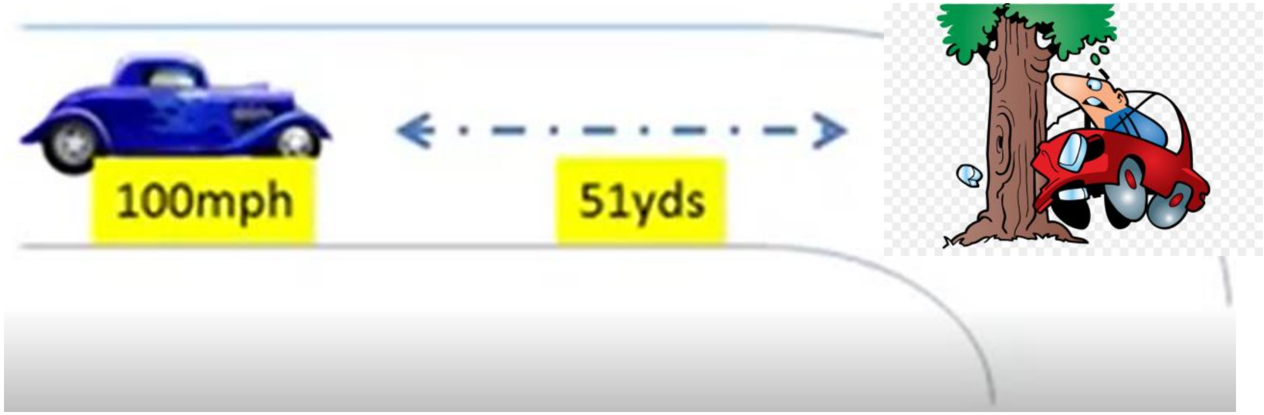
- Example:
  - Performance index: Model a sine wave
  - DOF: 1st order polynomial



- Model a sine wave with a 1st order polynomial!

DOF optimization needs long prediction horizon

- Example:
  - Performance index: Take a corner for the car in 51 yds.
  - DOF: Control speed and direction
  - Prediction horizon: 50 yds.



- Take a corner for the car in 51 yds with 50 yds prediction horizon!

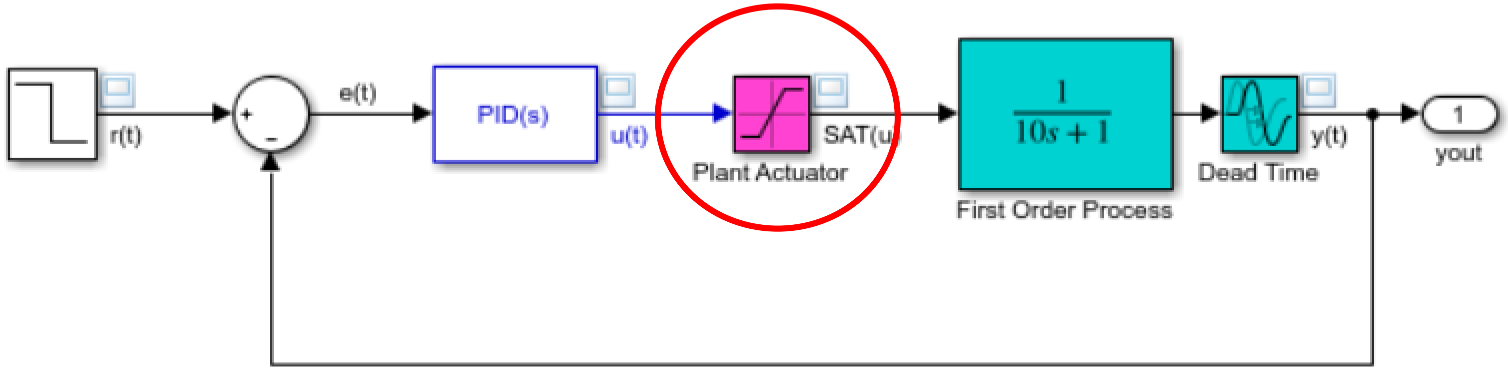
Main components of MPC

- Prediction
- Receding horizon
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

- More typical control strategies treat constraints as an **after thought**.

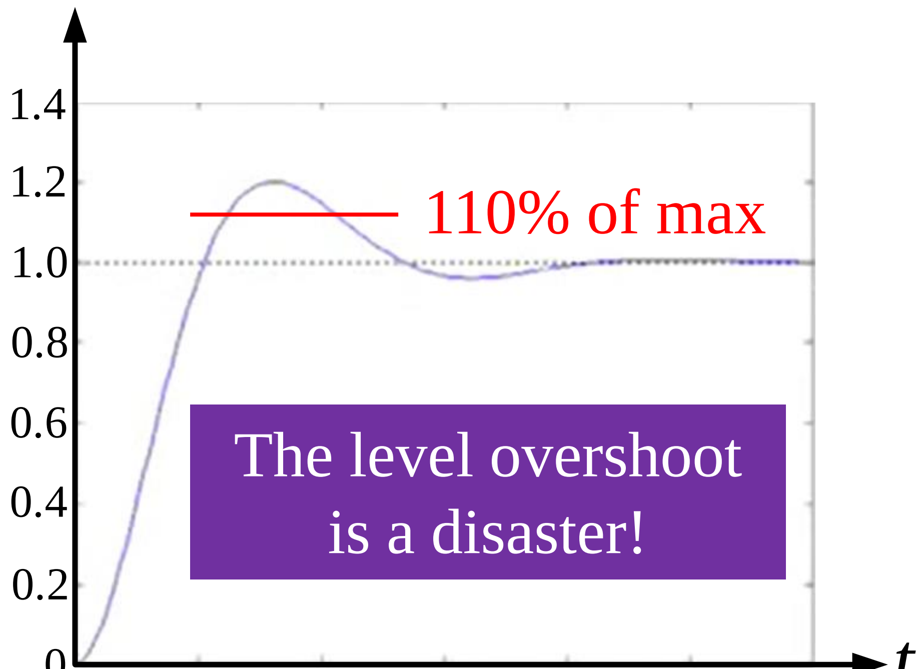


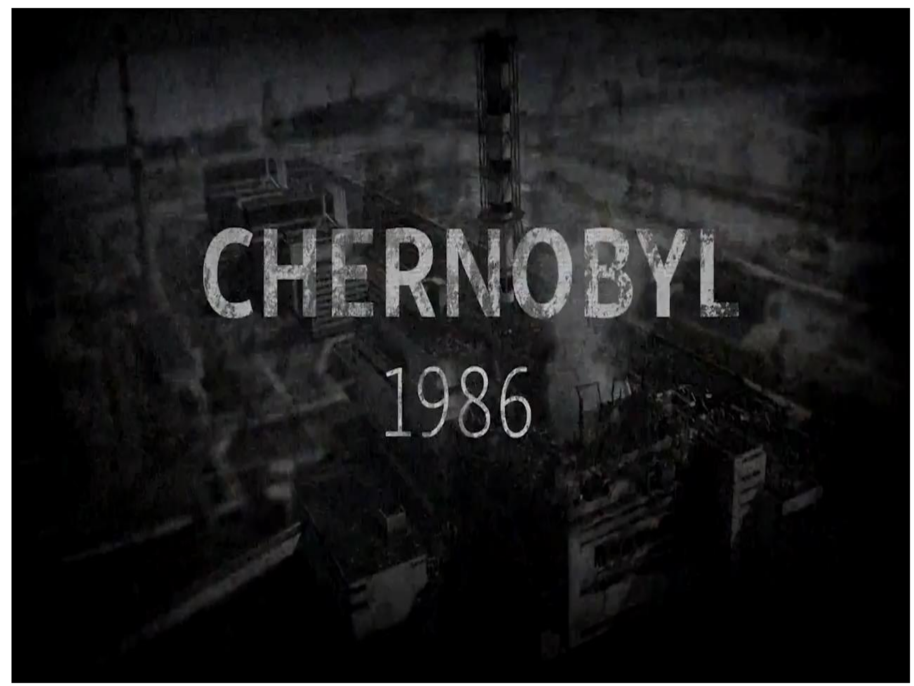
- MPC treat constraints as **a before thought:**
  - Embedded constraints into the strategy development.
  - The control action is optimal while satisfying constraints

Constrains is one of the advantage of MPC

Disaster of constraints problem

- PI control design: allow up to a 25% overshoot.
- Nuclear reactor: maximal permission is 110%.





MPC can fix the constrains problem

- Input flow of MPC: Not allow the tank to overflow. May result in slightly slower transients rise times, but **safe**.

- Input flow of MPC: Limited to 100% and avoid the instability problem cased by earlier input choices.

Embedding constrains can ensure the proposed MPC are optimized and safe for different operating points.

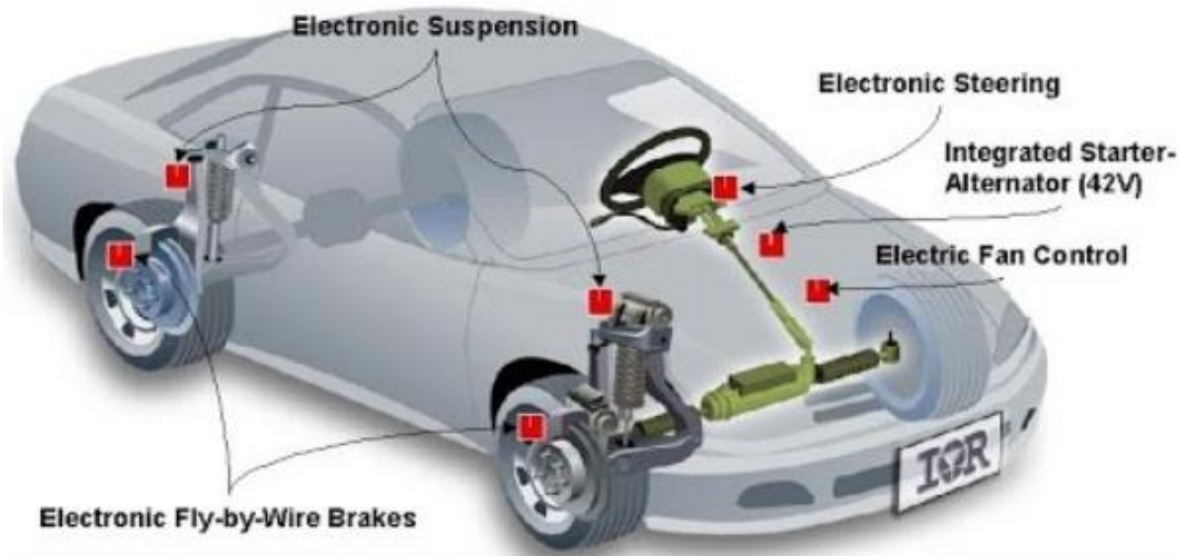
Main components of MPC

- Prediction
- Receding horizon
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

How MPC works?

Should **not** attempt MPC design **before** we have the required understanding

- Example car:
  - Inputs: throttle and steering
  - Outputs: speed and direction



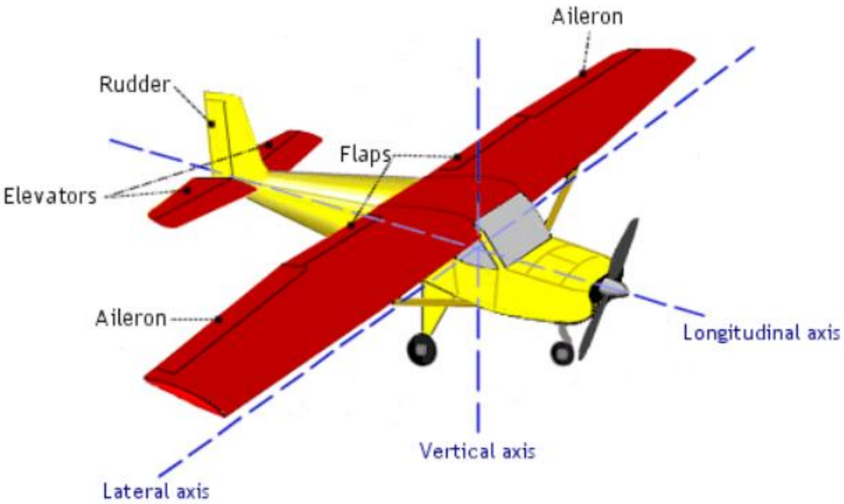
Multivariable processes: changing one input changes all the outputs

↓

Effective control law has to consider all inputs and outputs

Multivariable: Airplane

- Example airplane:
  - Inputs: numerous control surfaces
  - Outputs: moves in 3D space.



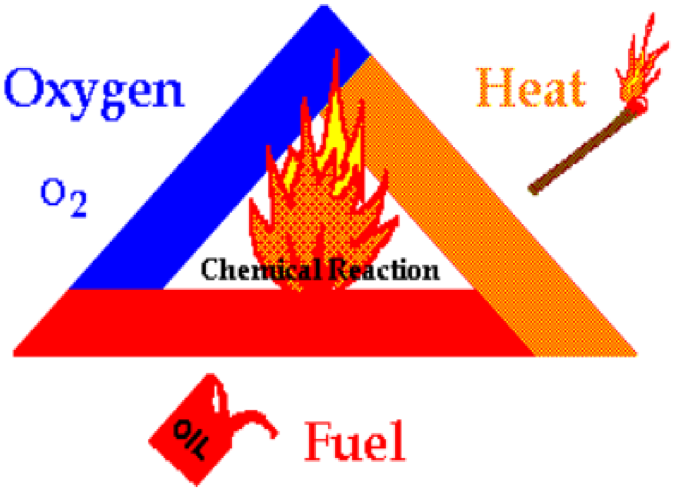
Multivariable processes: changing one input changes all the outputs

↓

Effective control law has to consider all inputs and outputs

Multivariable: Chemical reaction

- Example chemical reaction:
  - Inputs: Flow rates, heat supply, pressure, etc.
  - Outputs: Speed of reaction (production rate), quality, purity, etc.



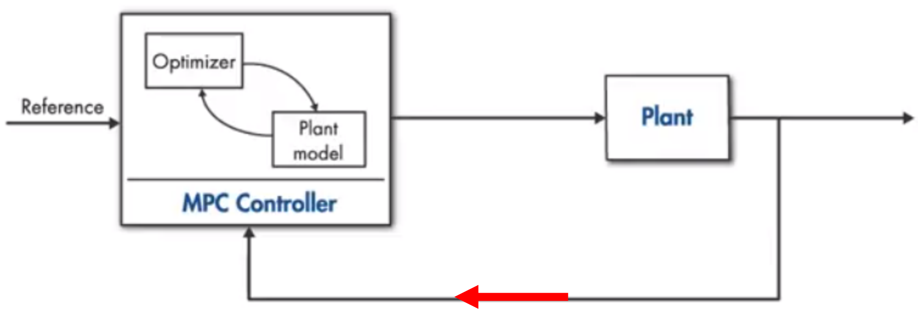
Multivariable processes: changing one input changes all the outputs

↓

Effective control law has to consider all inputs and outputs

Handing interaction between multivariables

- We can cope up to 2-3 inputs/outputs
- Beyond that a human is not an ideal controller!



Main components of MPC

- Prediction
- Receding horizon
- Modeling
- Performance index
- Degrees of freedom
- Constraint handling
- Multivariable

Should **not** attempt MPC design **before** we have the required understanding

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

BASIC THEORY OF THE MODEL PREDICTIVE CONTROL (MPC): PART I

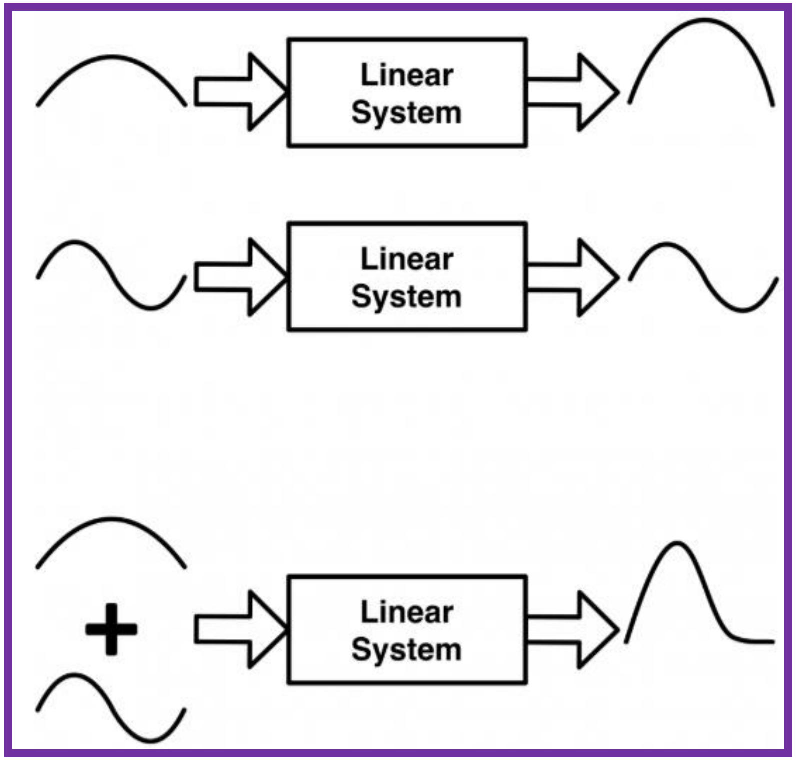
[17/10/2019]

- Main components of MPC
- Modelling of MPC
- MPC with state space model
- MPC with Carima model

MPC suitable for linear model

These slides do not discuss non-linear models

- Manipulation and algebra requires linear models as superposition can be used.

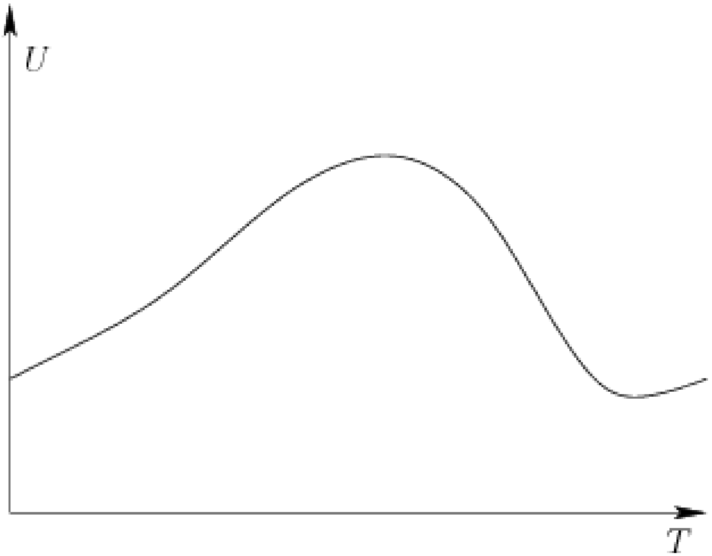


- Linear models are good enough for MPC.

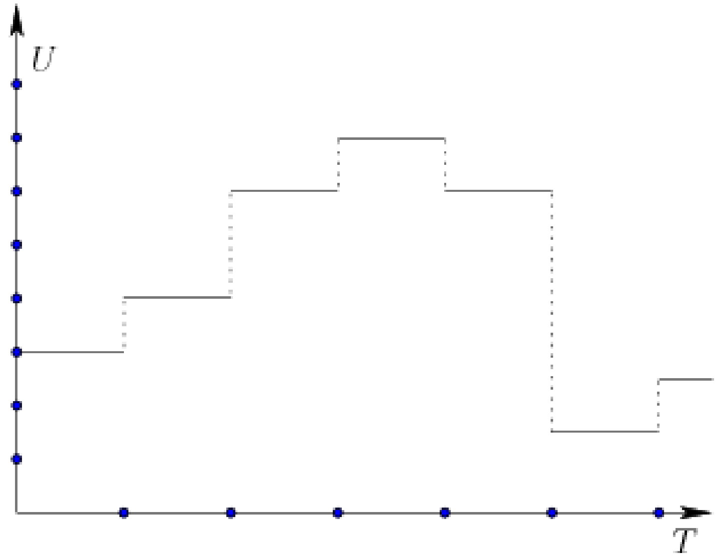
- Typical linear models:
  - Transfer function
  - State-space
  - Step response models (subset of transfer functions).

MPC is discrete or continuous?

- Processes operate in continuous time.
- MPC's Decision requires processing time and cannot instantaneous.



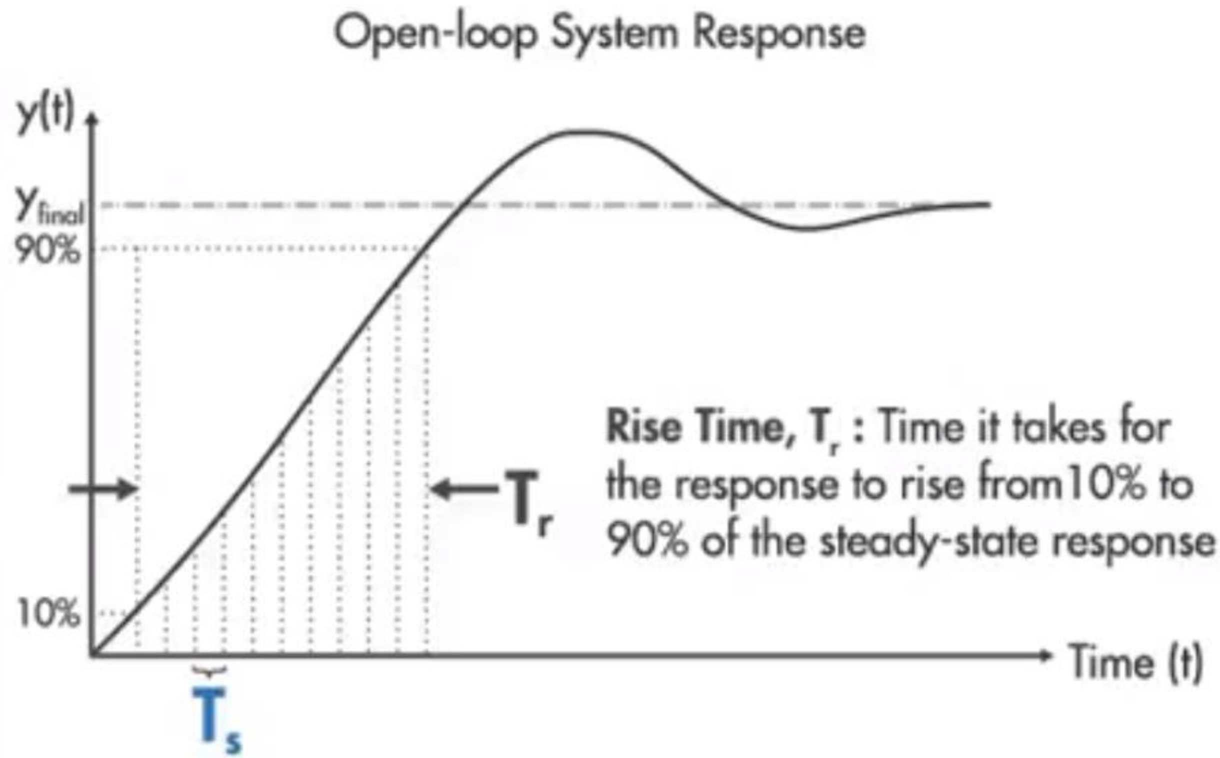
A trajectory in $\mathcal{U}$



A trajectory in $\mathcal{U}_d$

MPC laws are implemented in discrete time.

MPC needs suitable sample rate

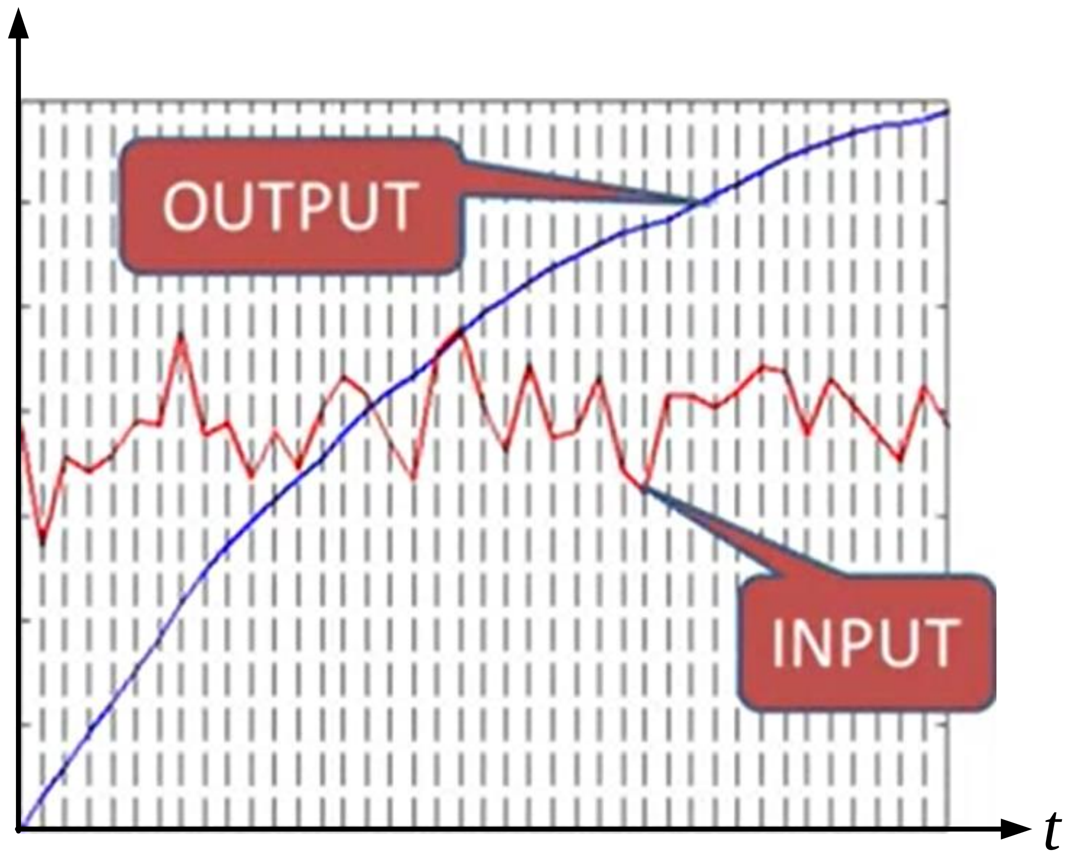


Sample Time, T_s: Controller execution rate

$$\frac{T_r}{20} \leqslant T_s \leqslant \frac{T_r}{10}$$

The fast sample rate is not good

High sample rate is pointless: System cannot respond to it



STATE SPACE MODEL

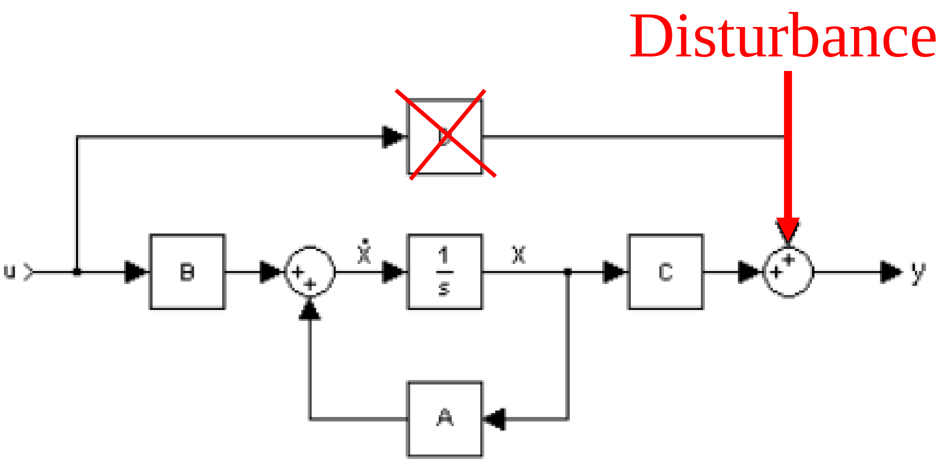
- Discrete state space model:

$$x_{k+1} = Ax_k + Bu_k$$

State Input

$$y_k = Cx_k + \cancel{Du_k} + d_k$$

Output Disturbance



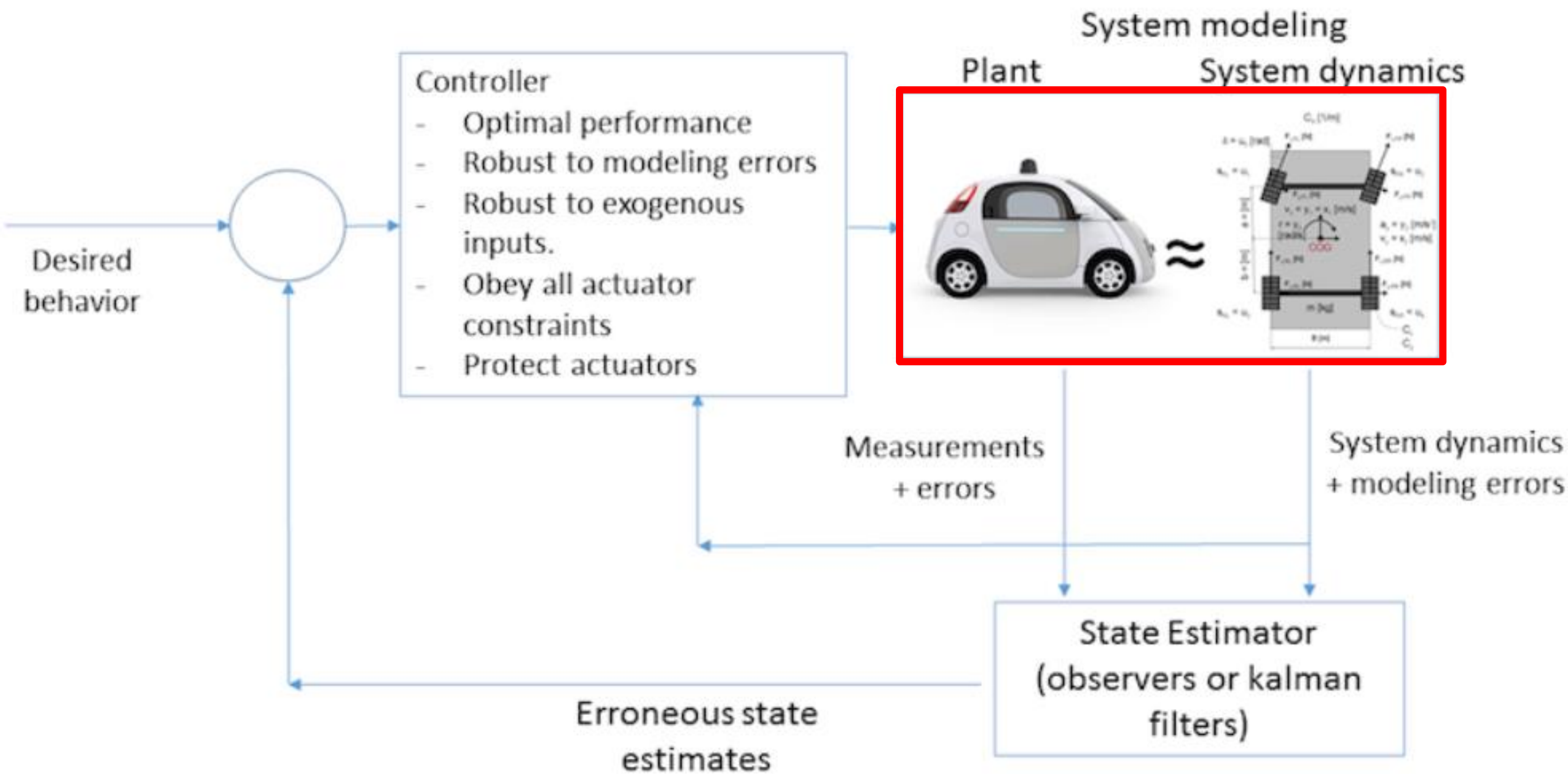
- In this slides, D=0 and add disturbance d_k.

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k + d_k$$

Application of state space model

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k + Du_k + d_k$$



TRANSFER FUNCTION MODEL

- Transfer function model with MPC is so called CARIMA Model.

$$a(z)y_k = b(z)u_k + T(z)\frac{\varsigma_k}{\Delta}$$

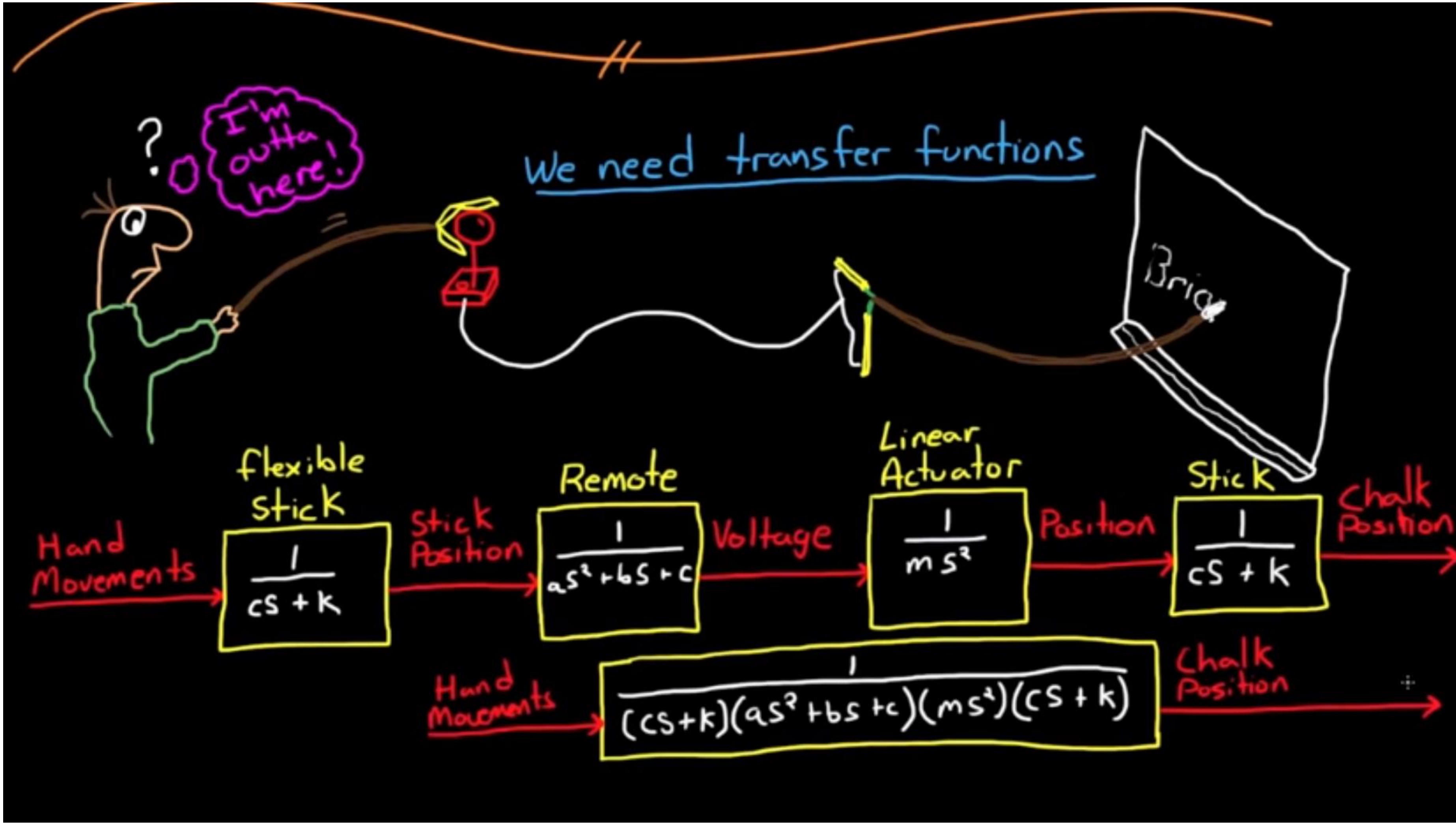
Output

Input

Disturbance

- Uncertainty is included
- Slowly varying disturbances is considered.

Application of transfer function model



STEP RESPONSE MODEL

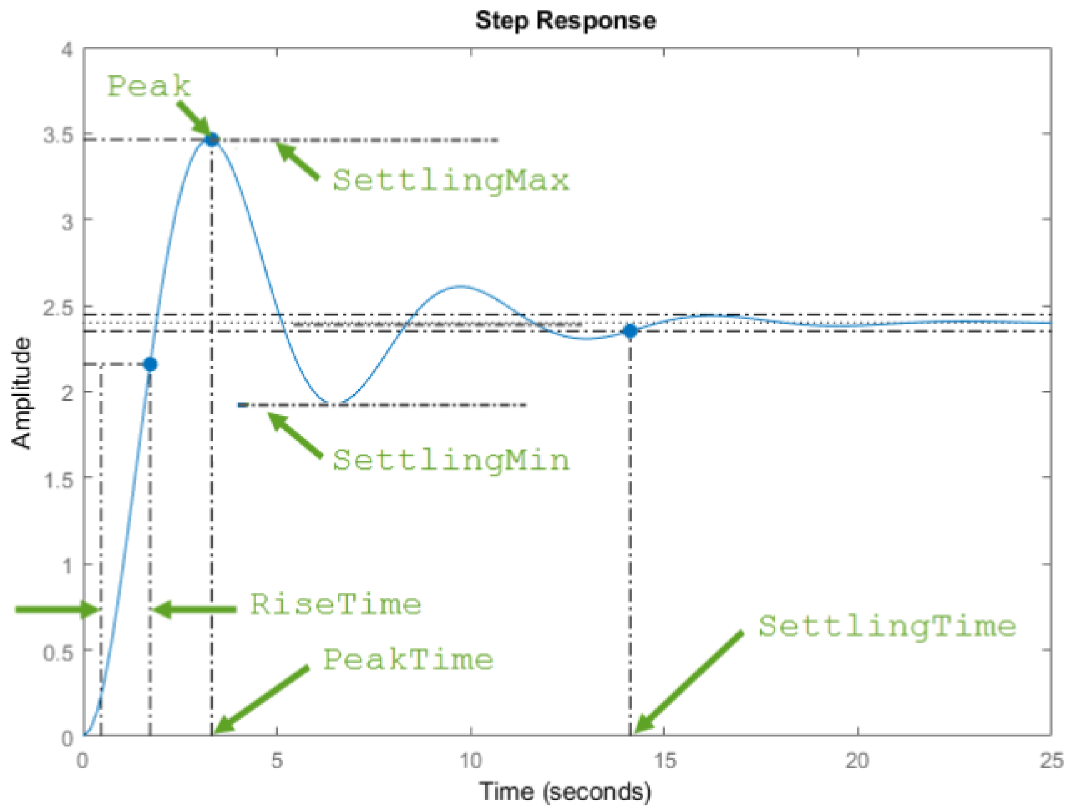
- Step response model is transfer function model.

$$y_k = H(z)\Delta u_k + d_k$$

Input

Output

Disturbance



- **Popular**: available characteristic for many process systems.
- **Disturbance**: difference between model output & measured output.

Summary

- MPC model is linear model.
- MPC is discrete model
- Faster sample rate or slow sample rate are not good
- State space model
- Transfer function model
- Step response model

BASIC THEORY OF THE MODEL PREDICTIVE CONTROL (MPC): PART I

[17/10/2019]

- Main components of MPC
- Modelling of MPC
- MPC with state space model
- MPC with Carima model

- Discrete state space model:

$$x_{k+1} = Ax_k + Bu_k$$

$$y_k = Cx_k + \cancel{Du_k} + d_k$$

State · Input · Output · Disturbance

- Assume 1: $D=0$.
- Assume 2:
  - n_x states for $\boldsymbol{x}$;
  - m_u inputs for $\boldsymbol{u}$
  - m_y outputs for $\boldsymbol{y}$.

We need these assumes

- One-step ahead prediction models,
  - Given data at sample 'k',
  - Determine data at sample '$k+1$'.

$$\left.\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + d_k \end{aligned}\right\} \Rightarrow \begin{aligned} y_{k+1} &= Cx_{k+1} + d_{k+1} \\ &\Downarrow \\ y_{k+1} &= CAx_k + CBu_k + d_{k+1} \end{aligned}$$

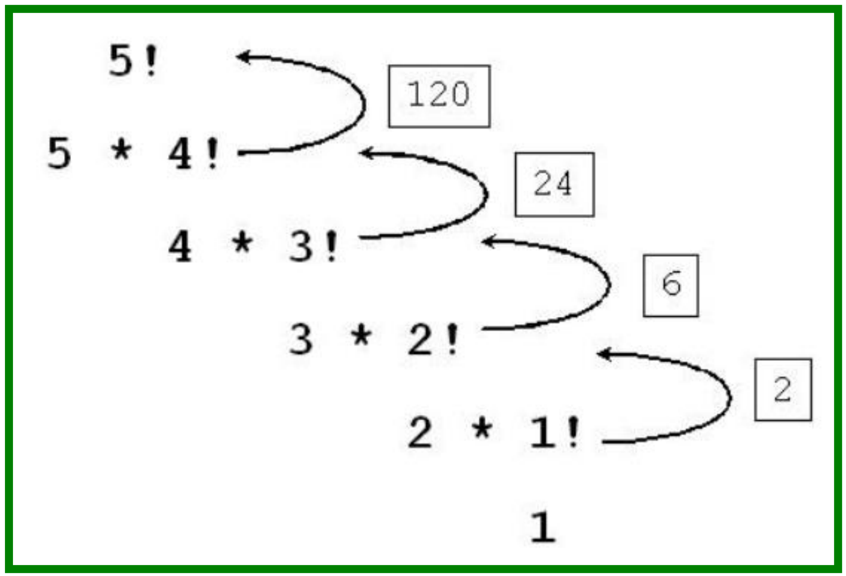
- Assume of disturbance:

$$d_k = d_{k+1}$$

$$\Downarrow$$

$$y_{k+1} = CAx_k + CBu_k + d_k$$

- One-step prediction can find *n*-step prediction recursively:



$$\left.\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ x_{k+2} &= Ax_{k+1} + Bu_{k+1} \\ x_{k+3} &= Ax_{k+2} + Bu_{k+2} \\ x_{k+4} &= Ax_{k+3} + Bu_{k+3} \end{aligned}\right\} \Rightarrow$$

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ x_{k+2} &= A\left(Ax_k + Bu_k\right) + Bu_{k+1} \\ x_{k+3} &= A\left[A\left(Ax_k + Bu_k\right) + Bu_{k+1}\right] + Bu_{k+2} \\ x_{k+4} &= A\left\{A\left[A\left(Ax_k + Bu_k\right) + Bu_{k+1}\right] + Bu_{k+2}\right\} + Bu_{k+3} \end{aligned}$$

Expanding prediction of x

$$\boxed{x_{k+1} = Ax_k + Bu_k}$$

$$\boxed{x_{k+2} = A\left(Ax_k + Bu_k\right) + Bu_{k+1}}$$

$$\boxed{x_{k+3} = A\left[\left(Ax_k + Bu_k\right) + Bu_{k+1}\right] + Bu_{k+2}}$$

$$\boxed{x_{k+4} = A\left\{\left[\left(Ax_k + Bu_k\right) + Bu_{k+1}\right] + Bu_{k+2}\right\} + Bu_{k+3}}$$

- Expanding the out:

$$\boxed{x_{k+1} = Ax_k + Bu_k}$$

$$\boxed{x_{k+2} = A^2 x_k + ABu_k + Bu_{k+1}}$$

$$\boxed{x_{k+3} = A^3 x_k + A^2 Bu_k + ABu_{k+1} + Bu_{k+2}}$$

$$\boxed{x_{k+4} = A^4 x_k + A^3 Bu_k + A^2 Bu_{k+1} + ABu_{k+2} + Bu_{k+3}}$$

The pattern is obvious.

General prediction formula

- n-step ahead prediction of x is:

$$x_{k+n} = A^n x_k + A^{n-1} B u_k + A^{n-2} B u_{k+1} + \ldots + A^1 B u_{k+n-2} + A^0 B u_{k+n-1}$$

- n-step ahead prediction of y is:

$$y_{k+n} = C x_{k+n} + d_{k+n} \qquad d_k = d_{k+n}$$

$$y_{k+n} = C\left(A^n x_k + A^{n-1} B u_k + A^{n-2} B u_{k+1} + \ldots + A^1 B u_{k+n-2} + A^0 B u_{k+n-1}\right) + d_{k+n}$$

$$y_{k+n} = C A^n x_k + C\left(A^{n-1} B u_k + A^{n-2} B u_{k+1} + \ldots + A^1 B u_{k+n-2} + A^0 B u_{k+n-1}\right) + d_k$$

- Mixed up **past** and **future** data;
- Be careful with **notation;**
- Be careful with **predictions** construction;

SIMPLIFICATION 1: DOUBLE SUBSCRIPT

Prediction notation: Double subscript

- Double subscript:
  - the 1st term: sample of the prediction (how many steps ahead);
  - the 2nd term: sample at which the prediction was made.

- Example:

$$x_{k+4|k}$$

Prediction of x at sample (k+4) where prediction was made at sample (k)

$$y_{k+6|k+2}$$

Prediction of y at sample (k+6) where prediction was made at sample (k+2)

Application of the double subscript

- Expression of the *n*-step ahead prediction is:

$$x_{k+n} = A^n x_k + A^{n-1}Bu_k + A^{n-2}Bu_{k+1} + \ldots + A^1 Bu_{k+n\text{-}2} + A^0 Bu_{k+n-1}$$

$$\Downarrow \text{ Double subscript}$$

$$x_{k+n|k} = A^n x_{k|k} + A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n-1|k}$$

$$y_{k+n} = CA^n x_k + C\left(A^{n-1}Bu_k + A^{n-2}Bu_{k+1} + \ldots + A^1 Bu_{k+n\text{-}2} + A^0 Bu_{k+n-1}\right) + d_k$$

$$\Downarrow \text{ Double subscript}$$

$$y_{k+n|k} = CA^n x_{k|k} + C\left(A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n-1|k}\right) + d_k$$

- Double subscript: a value is ‘in the future’ as opposed to known.

Splitting unknown and known parts

- Separate predictions into known and unknown parts (convenient).

$$y_{k+n|k} = CA^n x_{k|k} + C\left(A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n-1|k}\right) + d_k$$

$$\Downarrow$$

$$y_{k+n|k} = \left(CA^n x_{k|k} + d_k\right) + \left[C\left(A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n-1|k}\right)\right]$$

Known based on the **current** and **past** measurement

Unknown as based on the **future** input choices which remain to be decided

Aim: **choose 'unknown' inputs** to ensure **prediction is satisfactory**.

SIMPLIFICATION 2: VECTOR OF VECTORS

Vector of vectors notation

- Vector of vectors: A simple arrow notation captures a set of predictions.

$$\begin{bmatrix} x_{k+1|k} \\ x_{k+2|k} \\ \vdots \\ x_{k+n|k} \end{bmatrix} = \underset{\rightarrow}{x}_{k+1}$$

Subscript $k+1$ indicates the 1^{st} value in **the prediction vector**

Arrow pointing **right** means **prediction**

Applications of vector of vectors

- Using the 'arrow' notation:

$$\underset{\rightarrow}{x}_{k+1} = \begin{bmatrix} x_{k+1|k} \\ x_{k+2|k} \\ \vdots \\ x_{k+n|k} \end{bmatrix} = \begin{bmatrix} Ax_k + Bu_{k|k} \\ A^2 x_k + ABu_{k|k} + Bu_{k+1|k} \\ \vdots \\ A^n x_k + A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n\text{-}1|k} \end{bmatrix}$$

- Separating into past and decision variables gives:

$$\underset{\rightarrow}{x}_{k+1} = \begin{bmatrix} Ax_k \\ A^2 x_k \\ \vdots \\ A^n x_k \end{bmatrix} + \begin{bmatrix} Bu_{k|k} \\ ABu_{k|k} + Bu_{k+1|k} \\ \vdots \\ A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n\text{-}1|k} \end{bmatrix}$$

SIMPLIFICATION EXAMPLE

Combine predictions and notation

- Using matrix multiplication:

$$\underset{\rightarrow}{x}_{k+1} = \begin{bmatrix} Ax_k \\ A^2 x_k \\ \vdots \\ A^n x_k \end{bmatrix} + \begin{bmatrix} Bu_{k|k} \\ ABu_{k|k} + Bu_{k+1|k} \\ \vdots \\ A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n\text{-}1|k} \end{bmatrix}$$

A, B are model / prediction parameters

$$\underset{\rightarrow}{x}_{k+1} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix} \cdot x_k + \begin{bmatrix} B & 0 & \ldots & 0 \\ AB & B & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1}B & A^{n-2}B & \ldots & B \end{bmatrix} \cdot \begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ \vdots \\ {}_{k+n\text{-}1|k} \end{bmatrix}$$

Decision variables

Compact prediction notation

- Giving compact names:

$$\underset{\rightarrow}{x}_{k+1} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix}}_{P_x} \cdot x_k + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{n-1}B & A^{n-2}B & \cdots & B \end{bmatrix}}_{H_x} \cdot \underbrace{\begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ \vdots \\ k+n\text{-}1|k \end{bmatrix}}_{\underset{\rightarrow}{u}_k} \quad \underset{\rightarrow}{u}_k$$

$$\underset{\rightarrow}{x}_{k+1} = P_x \cdot x_k + H_x \cdot \underset{\rightarrow}{u}_k$$

Depends on **past**

Depends upon **decision variables**

Compact output prediction notation

- Output predictions follow a similar method.

$$y_{k+n|k} = \left(CA^n x_{k|k} + d_k\right) + \left[C\left(A^{n-1}Bu_{k|k} + A^{n-2}Bu_{k+1|k} + \ldots + ABu_{k+n\text{-}2|k} + Bu_{k+n-1|k}\right)\right]$$

$$\underset{\to}{y}_{k+1} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^n \end{bmatrix}}_{P} \cdot x_k + \underbrace{\begin{bmatrix} CB & 0 & \ldots & 0 \\ CAB & CB & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{n-1}B & CA^{n-2}B & \ldots & CB \end{bmatrix}}_{H} \cdot \underbrace{\begin{bmatrix} u_{k|k} \\ u_{k+1|k} \\ \vdots \\ u_{k+n\text{-}1|k} \end{bmatrix}}_{\underset{\to}{u}_k} + \underbrace{\begin{bmatrix} d_k \\ d_k \\ \vdots \\ d_k \end{bmatrix}}_{Ld_k}$$

$$\underset{\to}{y}_{k+1} = \left(P \cdot x_k + Ld_k\right) + H \cdot \underset{\to}{u}_k$$

Depends on **past**

Depends upon **decision variables**

Summary

- The overall prediction is expressed in a simple way

$$\underset{\rightarrow}{x}_{k+1} = \boxed{P_x \cdot x_k} + \boxed{H_x \cdot \underset{\rightarrow}{u}_k}$$

$$\underset{\rightarrow}{y}_{k+1} = \boxed{\left(P \cdot x_k + L d_k\right)} + \boxed{H \cdot \underset{\rightarrow}{u}_k}$$

Choose the suitable control values

Depends **on past**

Depends upon **decision variables**

- State space model can use a compact form for all prediction horizons.
- Predictions separate into a known and decision variables parts.

BASIC THEORY OF THE MODEL PREDICTIVE CONTROL (MPC): PART I

[17/10/2019]

- Main components of MPC
- Modelling of MPC
- MPC with state space model
- MPC with Carima model

Please note

- Carima model is transfer function model.
- Only SISO system is considered

- Transfer function model with MPC is so called CARIMA Model.

Input

Disturbance

$$a(z)y_k = b(z)u_k + T(z)\frac{\varsigma_k}{\Delta}$$

Output

Zero-mean random variable

$$\Delta = 1 - Z^{-1}$$

- **Uncertainty** is included
- **Slowly varying disturbances** is considered
- $T(z)$ is treated as a design parameters

- One-step ahead prediction models: Given data at sample 'k', Determine data at sample '$k+1$'.

$$a(z)y_{k+1} = b(z)u_{k+1} + T(z)\frac{\varsigma_k}{\Delta} = b(z)u_{k+1} + d_k$$

$$a(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}$$

$$b(z) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}$$

$$\Downarrow$$

$$y_{k+1} + a_1 y_k + \cdots + a_n y_{k-n+1} = b_1 u_k + b_2 u_{k-1} + \cdots + b_m u_{k-m+1} + d_k$$

$$\Downarrow$$

$$y_{k+1} = b_1 u_k + b_2 u_{k-1} + \cdots + b_m u_{k-m+1} + d_k - a_1 y_k - \cdots - a_n y_{k-n+1}$$

This slides **will not use** the double subscript notation of $y_{k+1|k}$ as the meaning is already clear

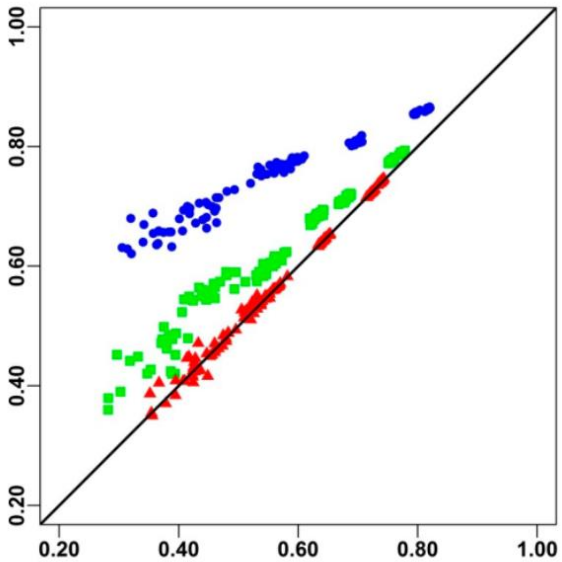
Significance of CARIMA model

- CARIMA model incorporates a disturbance estimate → can give unbiased predictions.

$$a(z)y_k = b(z)u_k + T(z)\frac{\varsigma_k}{\Delta}$$

$$\Downarrow$$

$$a(z)\Delta y_k = b(z)\Delta u_k + T(z)\varsigma_k$$



The incremental form is used for predictions
→
Based on changes rather than absolute values

Simplification of CARIMA model

- Simply: **assumes that the future ‘random’ term is zero**.

$$a(z)\Delta y_k = b(z)\Delta u_k + \cancel{T(z)\varsigma_k}$$

$$\left[a(z)\Delta\right] y_k = b(z)\left(\Delta u_k\right) \Rightarrow A(z)y_k = b(z)\Delta u_k$$

$$\boxed{A(z) = a(z)\Delta}$$

Combine a(z) and Delta

Use input increments

One step predication with simplified model

- One-step ahead prediction models: Given data at sample 'k', Determine data at sample '$k+1$'.

$$A(z)y_k = b(z)\Delta u_k$$

$$A(z) = a(z)\Delta$$

$$A(z) = 1 + A_1 z^{-1} + \cdots + A_n Z^{-n}$$

$$b(z) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}$$

$$y_{k+1} + A_1 y_k + \cdots + A_n y_{k-n+1} = b_1 \Delta u_k + b_2 \Delta u_{k-1} + \cdots + b_m \Delta u_{k-m+1}$$

$$y_{k+1} = b_1 \Delta u_k + b_2 \Delta u_{k-1} + \cdots + b_m \Delta u_{k-m+1} - A_1 y_k - \cdots - A_n y_{k-n+1}$$

No need for a **disturbance estimate** in this prediction model as within the use of **increments**

n step predication with simplified model

- *n*-step ahead prediction can be obtained by one-step ahead prediction with recursively:

$$y_{k+1} + A_1 y_k + \cdots + A_n y_{k-n+1} = b_1 \Delta u_k + b_2 \Delta u_{k-1} + \cdots + b_m \Delta u_{k-m+1}$$

$$y_{k+2} + A_1 y_{k+1} + \cdots + A_n y_{k-n+2} = b_1 \Delta u_{k+1} + b_2 \Delta u_k + \cdots + b_m \Delta u_{k-m+2}$$

$$y_{k+3} + A_1 y_{k+2} + \cdots + A_n y_{k-n+3} = b_1 \Delta u_{k+2} + b_2 \Delta u_{k+1} + \cdots + b_m \Delta u_{k-m+3}$$

$$y_{k+4} + A_1 y_{k+3} + \cdots + A_n y_{k-n+4} = b_1 \Delta u_{k+3} + b_2 \Delta u_{k+2} + \cdots + b_m \Delta u_{k-m+4}$$

  - Use the one step ahead to find y_{k+1},
  - Substitute y_{k+1} into the next equation to find y_{k+2},
  - Use y_{k+1} and y_{k+2} to find y_{k+3},
  - Keep iterating through to y_{k+n}.

Set of prediction equations

$$y_{k+1} + A_1 y_k + \cdots + A_n y_{k-n+1} = b_1 \Delta u_k + b_2 \Delta u_{k-1} + \cdots + b_m \Delta u_{k-m+1}$$

$$y_{k+2} + A_1 y_{k+1} + \cdots + A_n y_{k-n+2} = b_1 \Delta u_{k+1} + b_2 \Delta u_k + \cdots + b_m \Delta u_{k-m+2}$$

$$y_{k+3} + A_1 y_{k+2} + \cdots + A_n y_{k-n+3} = b_1 \Delta u_{k+2} + b_2 \Delta u_{k+1} + \cdots + b_m \Delta u_{k-m+3}$$

$$y_{k+4} + A_1 y_{k+3} + \cdots + A_n y_{k-n+4} = b_1 \Delta u_{k+3} + b_2 \Delta u_{k+2} + \cdots + b_m \Delta u_{k-m+4}$$

There are <u>4 unknowns</u> and 4 equations → can solve.

Reshape the outputs expression

- Separates future and past variables for the outputs.

$$\begin{matrix} y_{k+1} + A_1 y_k + \cdots + A_n y_{k-n+1} \\ y_{k+2} + A_1 y_{k+1} + \cdots + A_n y_{k-n+2} \\ y_{k+3} + A_1 y_{k+2} + \cdots + A_n y_{k-n+3} \\ y_{k+4} + A_1 y_{k+3} + \cdots + A_n y_{k-n+4} \end{matrix} = \underbrace{C_A \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ y_{k+4} \end{bmatrix}}_{\text{Future}} + \underbrace{H_A \begin{bmatrix} y_k \\ y_{k\text{-}1} \\ \vdots \\ y_{k\text{-}n+1} \end{bmatrix}}_{\text{Past}}$$

$$C_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ A_1 & 1 & 0 & 0 \\ A_2 & A_1 & 1 & 0 \\ A_3 & A_2 & A_1 & 1 \end{bmatrix} \qquad H_A = \begin{bmatrix} A_1 & A_2 & \cdots & A_{n-4} & A_{n-3} & A_{n-2} & A_{n-1} & A_n \\ A_2 & A_3 & \cdots & A_{n-3} & A_{n-2} & A_{n-1} & A_n & 0 \\ A_3 & A_4 & \cdots & A_{n-2} & A_{n-1} & A_n & 0 & 0 \\ A_4 & A_5 & \cdots & A_{n-1} & A_n & 0 & 0 & 0 \end{bmatrix}$$

Set of prediction equations

$$y_{k+1} + A_1 y_k + \cdots + A_n y_{k-n+1} = b_1 \Delta u_k + b_2 \Delta u_{k-1} + \cdots + b_m \Delta u_{k-m+1}$$

$$y_{k+2} + A_1 y_{k+1} + \cdots + A_n y_{k-n+2} = b_1 \Delta u_{k+1} + b_2 \Delta u_k + \cdots + b_m \Delta u_{k-m+2}$$

$$y_{k+3} + A_1 y_{k+2} + \cdots + A_n y_{k-n+3} = b_1 \Delta u_{k+2} + b_2 \Delta u_{k+1} + \cdots + b_m \Delta u_{k-m+3}$$

$$y_{k+4} + A_1 y_{k+3} + \cdots + A_n y_{k-n+4} = b_1 \Delta u_{k+3} + b_2 \Delta u_{k+2} + \cdots + b_m \Delta u_{k-m+4}$$

There are <u>4 unknowns</u> and 4 equations → can solve.

Reshape the control variables expression

- Separates future and past control variables.

$$
\begin{matrix}
b_1\Delta u_k + b_2\Delta u_{k-1} + \cdots + b_m\Delta u_{k-m+1} \\
b_1\Delta u_{k+1} + b_2\Delta u_k + \cdots + b_m\Delta u_{k-m+2} \\
b_1\Delta u_{k+2} + b_2\Delta u_{k+1} + \cdots + b_m\Delta u_{k-m+3} \\
b_1\Delta u_{k+3} + b_2\Delta u_{k+2} + \cdots + b_m\Delta u_{k-m+4}
\end{matrix}
= \underbrace{C_b \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \Delta u_{k+3} \end{bmatrix}}_{\text{Future}} + \underbrace{H_b \begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-m+1} \end{bmatrix}}_{\text{Past}}
$$

$$
C_b = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ b_2 & b_1 & 0 & 0 \\ b_3 & b_2 & b_1 & 0 \\ b_4 & b_3 & b_2 & b_1 \end{bmatrix}
\qquad
H_b = \begin{bmatrix} b_2 & b_3 & \cdots & b_{m-4} & b_{m-3} & b_{m-2} & b_{m-1} & b_m \\ b_3 & b_4 & \cdots & b_{m-3} & b_{m-2} & b_{m-1} & b_m & 0 \\ b_4 & b_5 & \cdots & b_{m-2} & b_{m-1} & b_m & 0 & 0 \\ b_5 & b_6 & \cdots & b_{m-1} & b_m & 0 & 0 & 0 \end{bmatrix}
$$

- Compact description of the entire predictions.

$$C_A \underbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ y_{k+4} \end{bmatrix}}_{\underset{\rightarrow}{y}_{k+1}} + H_A \underbrace{\begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-n+1} \end{bmatrix}}_{\underset{\leftarrow}{y}_{k}} = C_b \underbrace{\begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \Delta u_{k+2} \\ \Delta u_{k+3} \end{bmatrix}}_{\Delta \underset{\rightarrow}{u}_{k}} + H_b \underbrace{\begin{bmatrix} \Delta u_{k-1} \\ \Delta u_{k-2} \\ \vdots \\ \Delta u_{k-m+1} \end{bmatrix}}_{\Delta \underset{\leftarrow}{u}_{k-1}}$$

- Re-introduce the arrow notation as:

$$\boxed{C_A \cdot \underset{\rightarrow}{y}_{k+1} + H_A \cdot \underset{\leftarrow}{y}_{k} = C_b \cdot \Delta \underset{\rightarrow}{u}_{k} + H_b \cdot \Delta \underset{\leftarrow}{u}_{k-1}}$$

- Output predictions can be solved as:

$$\boxed{C_A \cdot \underset{\rightarrow}{y}_{k+1} + H_A \cdot \underset{\leftarrow}{y}_k = C_b \cdot \Delta \underset{\rightarrow}{u}_k + H_b \cdot \Delta \underset{\leftarrow}{u}_{k-1}}$$

$$\Downarrow$$

$$\boxed{\underset{\rightarrow}{y}_{k+1} = C_A^{-1} C_b \cdot \Delta \underset{\rightarrow}{u}_k + \left(C_A^{-1} H_b \cdot \Delta \underset{\leftarrow}{u}_{k-1} - C_A^{-1} H_A \cdot \underset{\leftarrow}{y}_k \right)}$$

Decision variables (**future input increments**)

Known or **past** data

- Simplify the expression.

$$\underset{\rightarrow}{y}_{k+1} = C_A^{-1}C_b \cdot \Delta \underset{\rightarrow}{u}_k + \left(C_A^{-1}H_b \cdot \Delta \underset{\leftarrow}{u}_{k-1} - C_A^{-1}H_A \cdot \underset{\leftarrow}{y}_k \right)$$

$$H = C_A^{-1}C_b \qquad P = C_A^{-1}H_b \qquad Q = C_A^{-1}H_A$$

$$\boxed{\underset{\rightarrow}{y}_{k+1} = H \cdot \Delta \underset{\rightarrow}{u}_k + \left(P \cdot \Delta \underset{\leftarrow}{u}_{k-1} - Q \cdot \underset{\leftarrow}{y}_k \right)}$$

Example 1

$$A(z)y_k = b(z)\Delta u_k \qquad a(z) = 1 - 0.8z^{-1} \qquad b(z) = 2z^{-1} + z^{-2}$$

- Using the definition of the prediction matrices and a horizon of 4.

$$\left.\begin{aligned} a(z) &= 1 - 0.8z^{-1} \\ \Delta &= 1 - Z^{-1} \end{aligned}\right\} \Rightarrow A(z) = a(z)\Delta = 1 \underset{A_1}{-1.8}z^{-1} + \underset{A_2}{0.8}z^{-2} + \underset{A_{3,4\cdots} = 0}{0}$$

$$C_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ A_1 & 1 & 0 & 0 \\ A_2 & A_1 & 1 & 0 \\ A_3 & A_2 & A_1 & 1 \end{bmatrix} \Rightarrow C_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1.8 & 1 & 0 & 0 \\ 0.8 & -1.8 & 1 & 0 \\ 0 & 0.8 & -1.8 & 1 \end{bmatrix}$$

$$H_A = \begin{bmatrix} A_1 & A_2 \\ A_2 & A_3 \\ A_3 & A_4 \\ A_4 & A_5 \end{bmatrix} \Rightarrow H_A = \begin{bmatrix} -1.8 & 0.8 \\ 0.8 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Example 1

$$A(z)y_k = b(z)\Delta u_k \qquad a(z) = 1 - 0.8z^{-1} \qquad b(z) = 2z^{-1} + z^{-2}$$

- Using the definition of the prediction matrices and a horizon of 4.

$$b(z) = 2z^{-1} + z^{-2} + 0 \Rightarrow b_1 = 2 \quad b_2 = 1 \quad b_{3,4\cdots} = 0$$

$$C_b = \begin{bmatrix} b_1 & 0 & 0 & 0 \\ b_2 & b_1 & 0 & 0 \\ b_3 & b_2 & b_1 & 0 \\ b_4 & b_3 & b_2 & b_1 \end{bmatrix} \Rightarrow C_b = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

$$H_b = \begin{bmatrix} b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \Rightarrow H_b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Example 1

$$A(z)y_k = b(z)\Delta u_k \qquad a(z) = 1 - 0.8z^{-1} \qquad b(z) = 2z^{-1} + z^{-2}$$

- Using the definition of the prediction matrices and a horizon of 4.

$$\boxed{\underset{\rightarrow}{y}_{k+1} = H \cdot \Delta \underset{\rightarrow}{u}_k + \left(P \cdot \Delta \underset{\leftarrow}{u}_{k-1} - Q \cdot \underset{\leftarrow}{y}_k \right)}$$

$$H = C_A^{-1} C_b \qquad P = C_A^{-1} H_b \qquad Q = C_A^{-1} H_A$$

$$C_A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1.8 & 1 & 0 & 0 \\ 0.8 & -1.8 & 1 & 0 \\ 0 & 0.8 & -1.8 & 1 \end{bmatrix} \qquad C_b = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

$$H_A = \begin{bmatrix} -1.8 & 0.8 \\ 0.8 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad H_b = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Review of today's topic

- Understand the concept of the MPC main components and their selection rules
- Understand the Modelling of the MPC
- Understand MPC with state space model
- Understand MPC with Carima model

Thank you!

Jackzhang@ntu.edu.sg