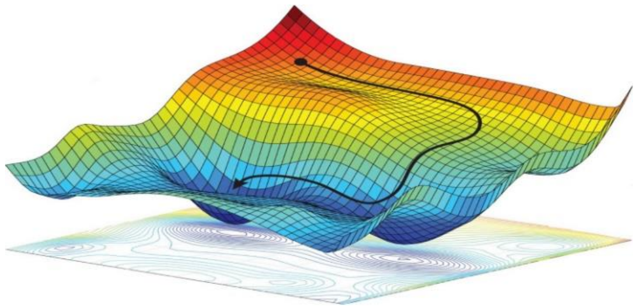
System Intelligence Laboratory

WONIK-KAIST FTC 기술 전수 세미나:
Decision-Making with Data-Driven Model

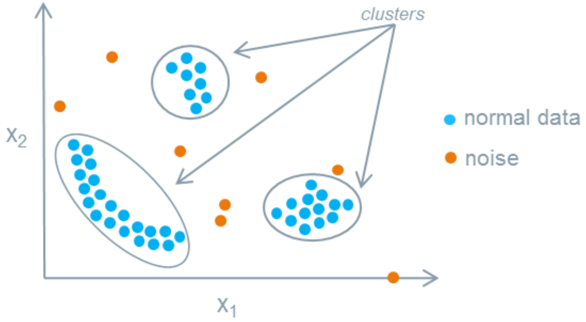
Dept. of Industrial & Systems engineering, KAIST
Chihyeon Song, Haewon Jung, Jinkyoo Park

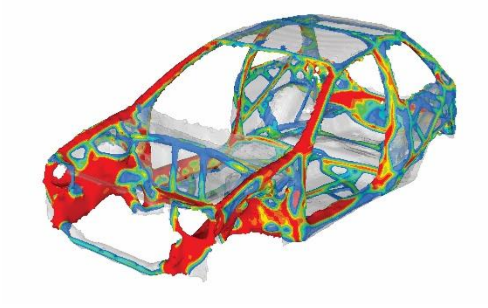
Decision-Making

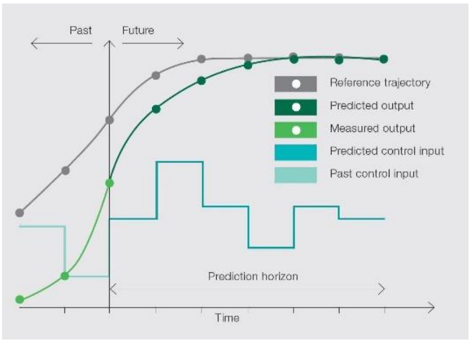
- Decision-making: What we really want to do

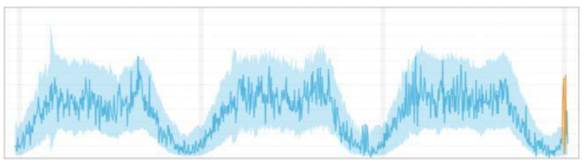


$$\min_{u_1,..,u_T} \sum_{t=1}^{T} c(x_t, u_t)$$
$$s.t. \quad x_{t+1} = f(x_t, u_t)$$



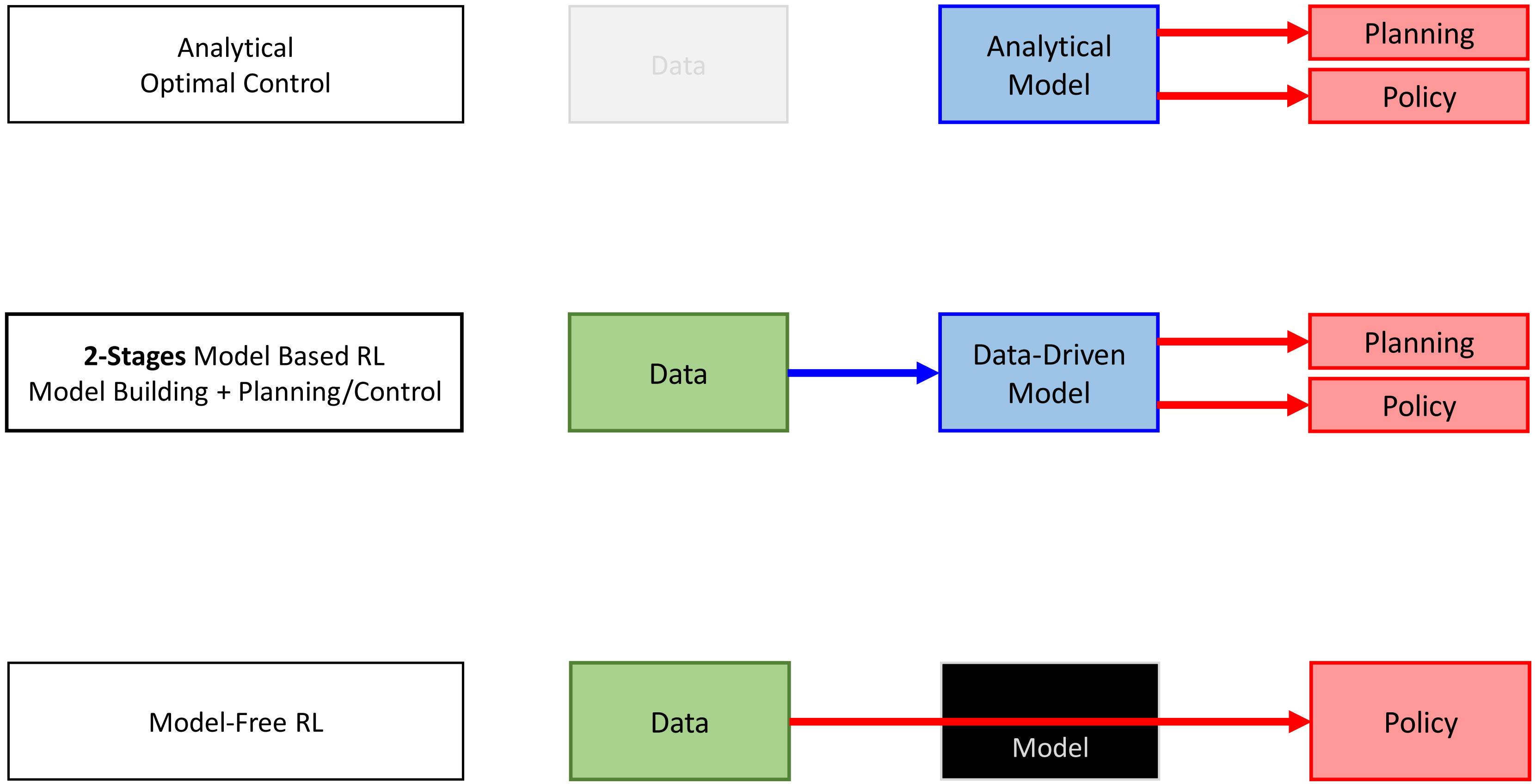






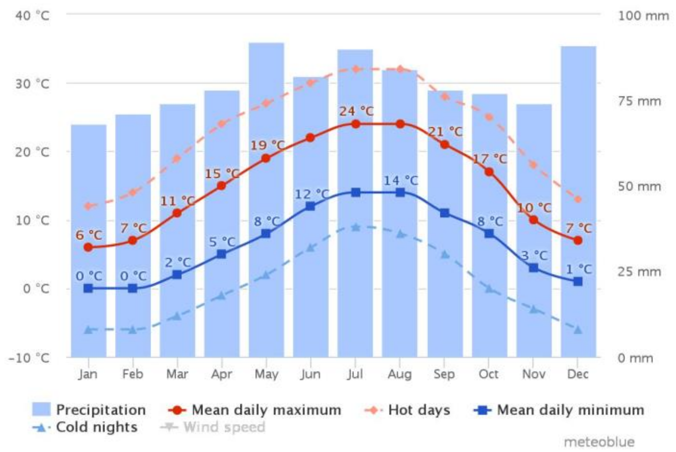
Model-based optimizations | **Optimal controls** | **Anomaly detections**

Ways to Make a Decision

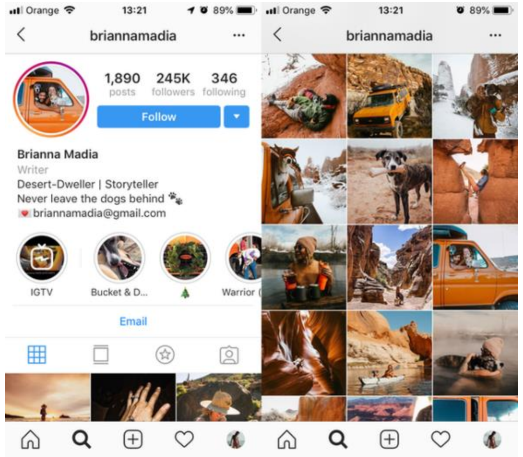


Decision-Making with Data-Driven Models

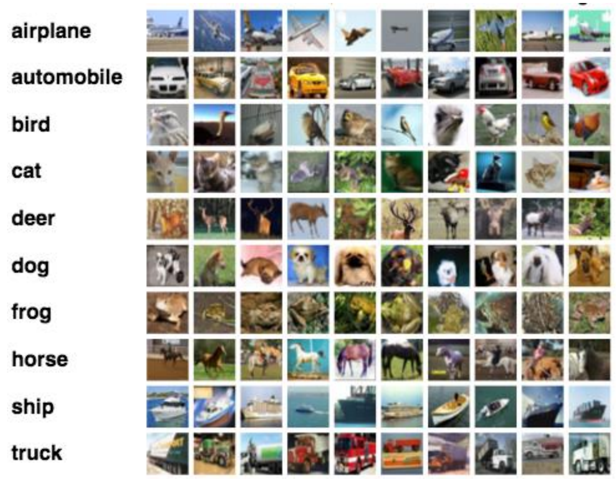
- Data: discrete value that contains information



Weather



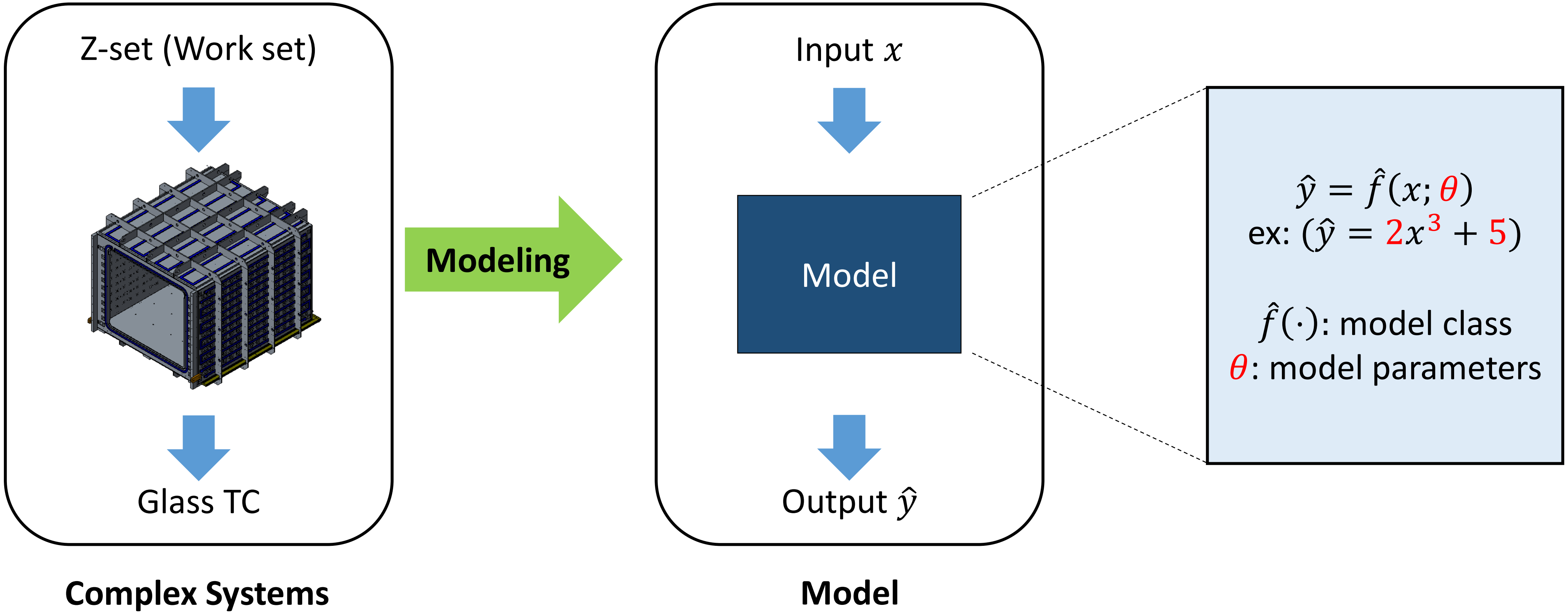
SNS



Image

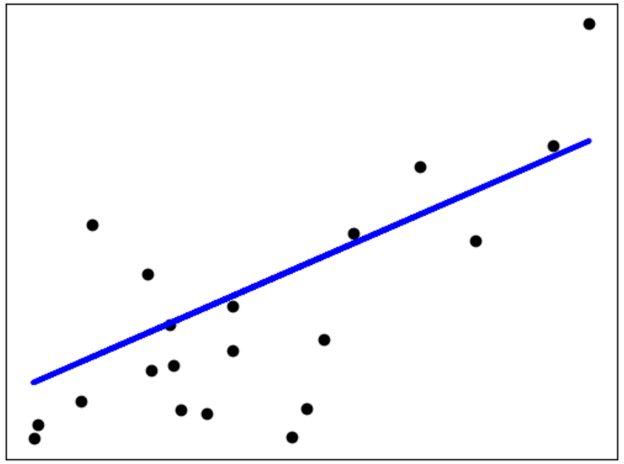
Decision-Making with Data-Driven Models

- Model: Theoretical representation of a system
  - Will be used in decision-making instead of the actual system

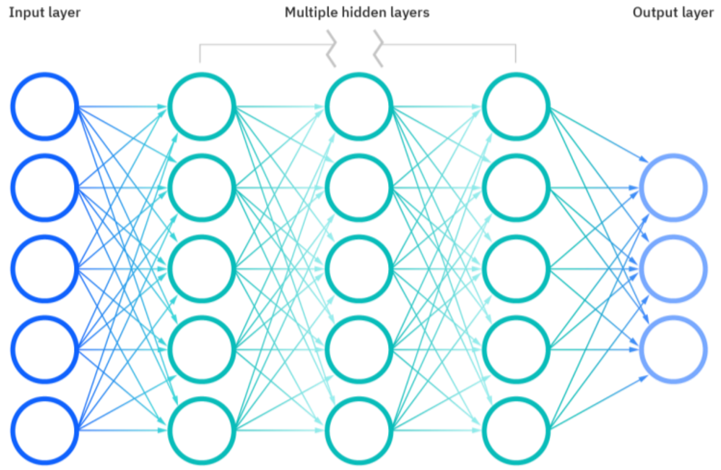


Decision-Making with Data-Driven Models

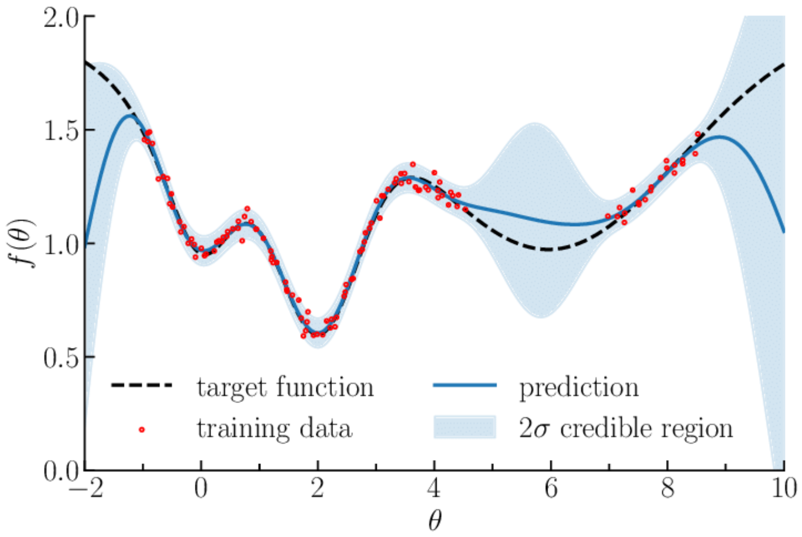
- Data-driven model: Model parameters are *learned* from the data
  - Collect the information from the data in terms of modeling



Linear model



Deep neural networks



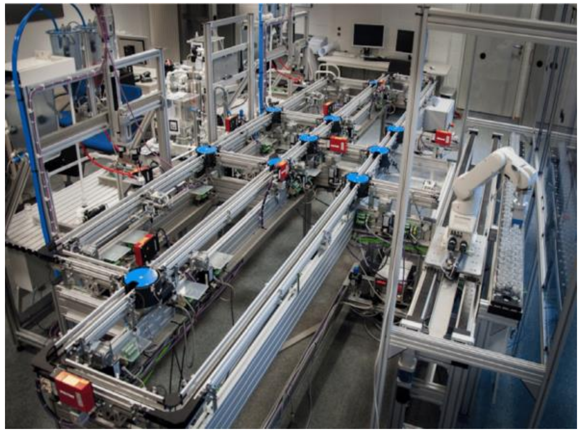
Gaussian process

Why Data-Driven Decision-Making?

- Complexity of modern systems
  - Data-driven approach doesn't require any background knowledge of the system

$$\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)+\frac{\partial}{\partial z}\left(k\frac{\partial T}{\partial z}\right)+q_V=\rho c_p\frac{\partial T}{\partial t}$$

where
k is the materials conductivity [W.m^{-1}.K^{-1}]
q_V is the rate at which energy is generated per unit volume of the medium [W.m^{-3}]
ρ is the density [kg.m^{3}]
c_p is the specific heat capacity [J.kg^{-1}.K^{-1}]



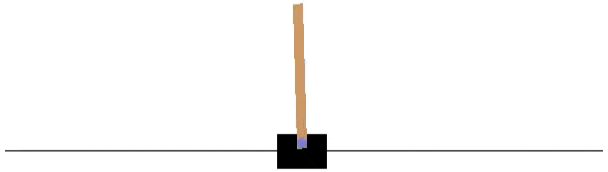
Scientific Knowledge

Modern Complex Systems

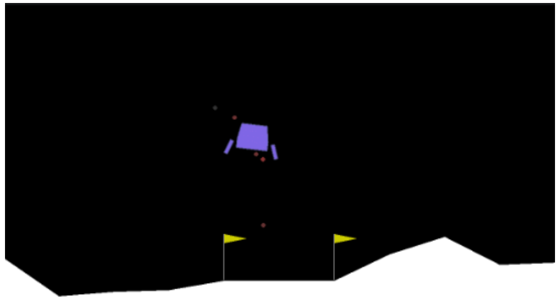
Why Data-Driven Model?

- In the real-world, we often have limited amount of data

- With an appropriate model, more data-efficient than model-free approach
  - More practical approach to the real-world systems

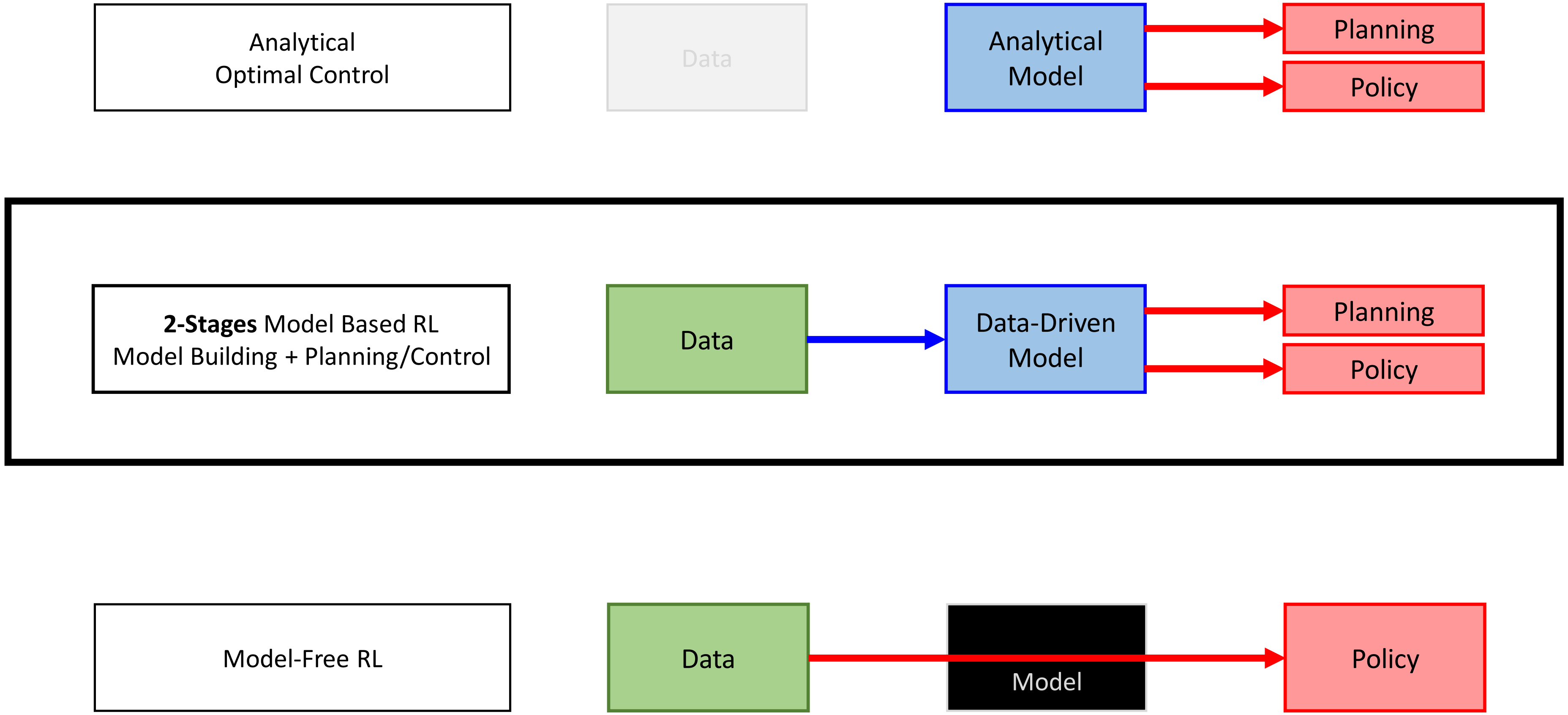


CartPole



LunarLander

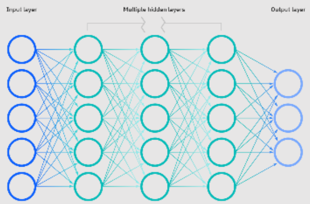
How to Make a Decision

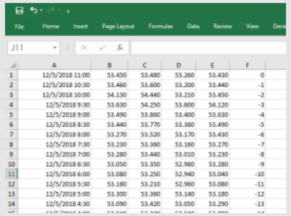


Two Big Questions

Q1. How we build data-driven model?

- Formulate System
- Model Selection
- Data Preprocessing
- Learn Model





Q2. How we make a decision with model?

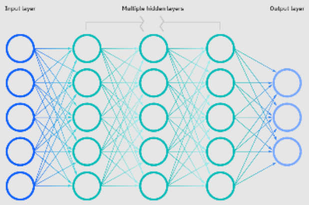
- Define Decision-Making
- Formulate Optimization Problem
- Solve Control Optimization
- Validate Control Performance

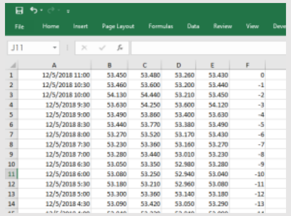
$$\min_x f(x) \quad s.t.\ g_i(x) \le 0\ \forall i = 1, \dots, I \quad h_j(x) = 0\ \forall j = 1, \dots, J$$

Two Big Questions

Q1. How we build data-driven model?

- Formulate System
- Model Selection
- Data Preprocessing
- Learn Model





	A	B	C	D	E	F
1	12/5/2018 11:00	53.450	53.480	53.260	53.430	0
2	12/5/2018 10:30	53.460	53.600	53.200	53.440	-1
3	12/5/2018 10:00	54.130	54.440	53.210	53.450	-2
4	12/5/2018 9:30	53.630	54.250	53.600	54.120	-3
5	12/5/2018 9:00	53.490	53.860	53.400	53.630	-4
6	12/5/2018 8:30	53.440	53.770	53.380	53.490	-5
7	12/5/2018 8:00	53.270	53.520	53.170	53.430	-6
8	12/5/2018 7:30	53.230	53.360	53.160	53.270	-7
9	12/5/2018 7:00	53.280	53.440	53.010	53.230	-8
10	12/5/2018 6:30	53.050	53.350	52.980	53.280	-9
11	12/5/2018 6:00	53.080	53.250	52.940	53.040	-10
12	12/5/2018 5:30	53.180	53.210	52.960	53.080	-11
13	12/5/2018 5:00	53.300	53.360	53.140	53.180	-12
14	12/5/2018 4:30	53.090	53.420	53.050	53.290	-13

Day 1

Q2. How we make a decision with model?

- Define Decision-Making
- Formulate Optimization Problem
- Solve Control Optimization
- Validate Control Performance

$$\min_{x} f(x)$$
$$s.t.\ g_i(x) \le 0\ \forall i = 1, \dots, I$$
$$h_j(x) = 0\ \forall j = 1, \dots, J$$

Day 2

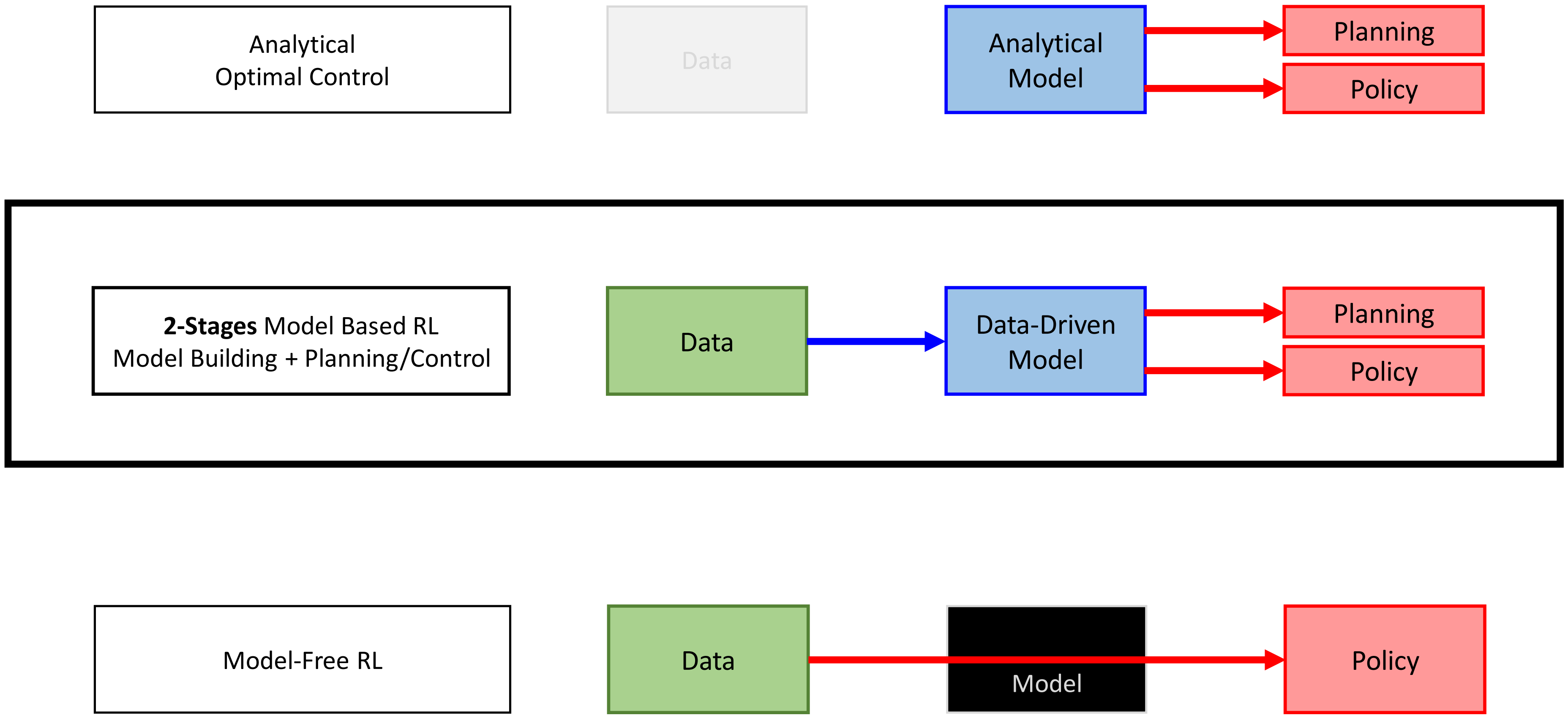
Q2. How We Make a Decision with Model?

We have a Model Now. And What?

- (Recap) Role of Model: Substitute the true system in decision-making

- With the model, we can 1) plan the future action or 2) learn a policy

How to Make a Decision

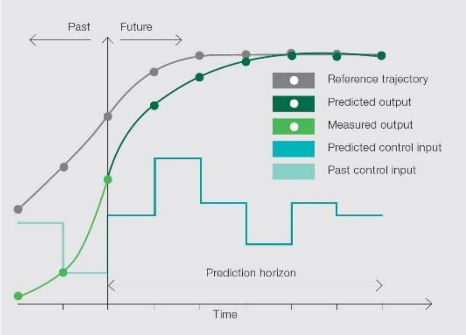


Planning vs Policy

$$\underset{u_{1:W}}{\operatorname{argmin}}\left\{\sum_{t=1}^{W}\gamma^t(\hat{x}_t - r_t)^2 + \lambda\sum_{t=1}^{W}(u_t - u_{t-1})^2\right\}$$

$$s.t.\ \hat{x}_{1:W} = f(x_0, u_{1:W})$$

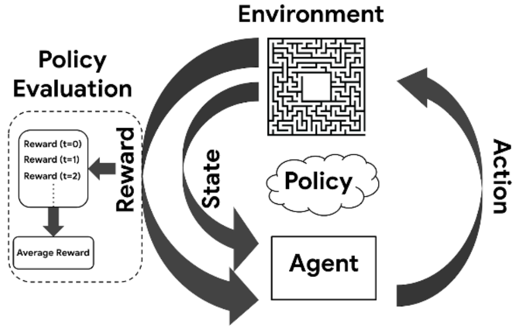
$$u_{min} \le u_{1:W} \le u_{max}$$



Theorem 1 (Policy Gradient). For any MDP, in either the average-reward or start-state formulations,

$$\frac{\partial\rho}{\partial\theta} = \sum_s d^\pi(s)\sum_a \frac{\partial\pi(s,a)}{\partial\theta}Q^\pi(s,a). \quad (2)$$

Proof: See the appendix.



Planning

$$u^* = \min_u \mathcal{L}(x, u)$$

Policy

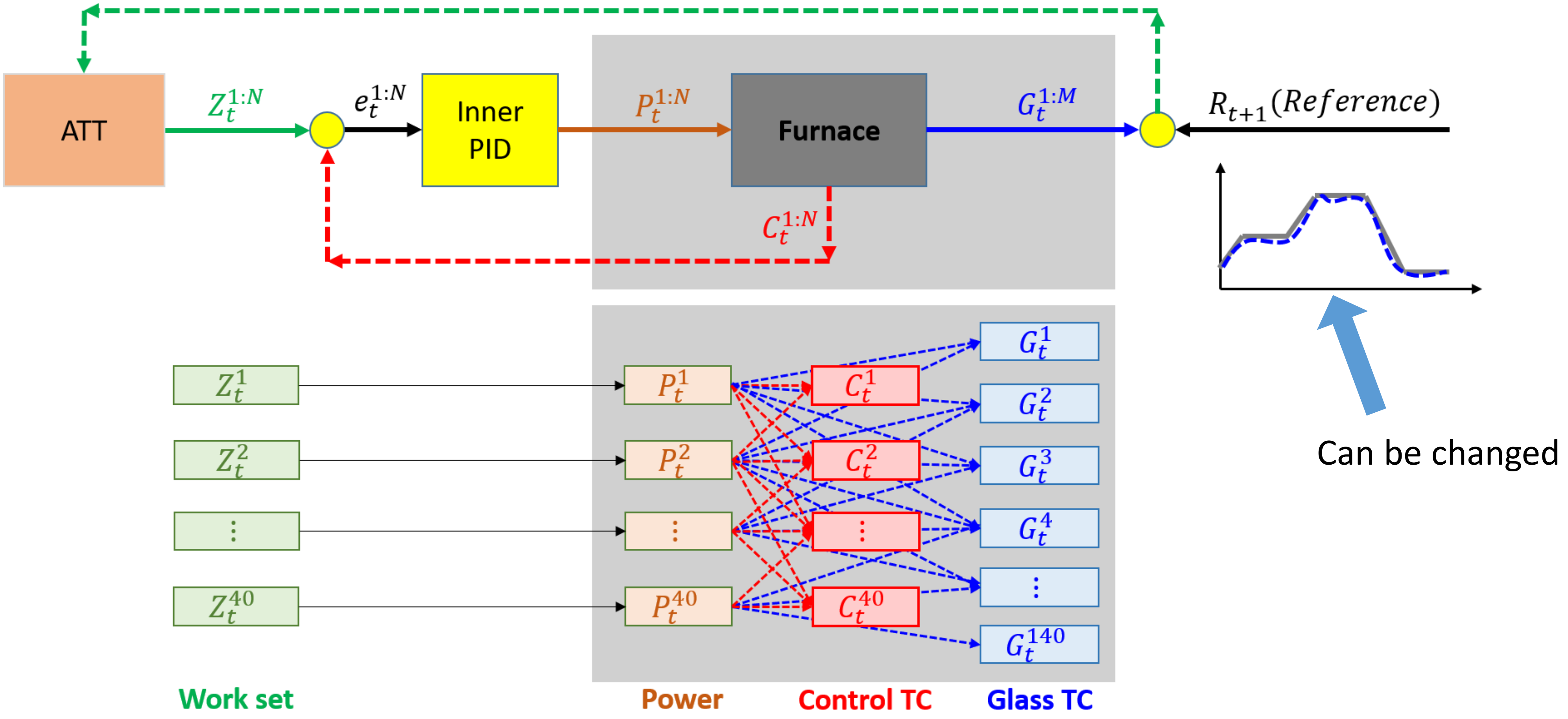
$$u^* = \pi(x_t)$$

Why Planning in FTC?

- Human can understand the decision-making logic
  - Decision is decided by solving optimization problem

- Easily adapt when the control goal is changed
  - Policy (often) assumes that the control goal is fixed

FTC Control Objective



Model Predictive Control (MPC)

모델 (Model)을 가지고 미래를 예측 (Predictive) 을 하면서, 제어 (Control) 하겠다.

$$\min_{\{u_\tau\}_{\tau=t}^{t+H-1}} \sum_{\tau=t+1}^{t+H} \|\hat{\boldsymbol{x}}_\tau - \boldsymbol{r}_\tau\|^2$$

$$\text{s.t.}\ \hat{x}_{\tau+1} = \hat{f}(x_\tau, u_\tau; \theta)$$

$$\Rightarrow$$

$$\{u_\tau\}_{\tau=t}^{t+H-1} \leftarrow \mathrm{MPC}(x_t, \{r_\tau\}_{\tau=t+1}^{t+H}; \hat{f})$$

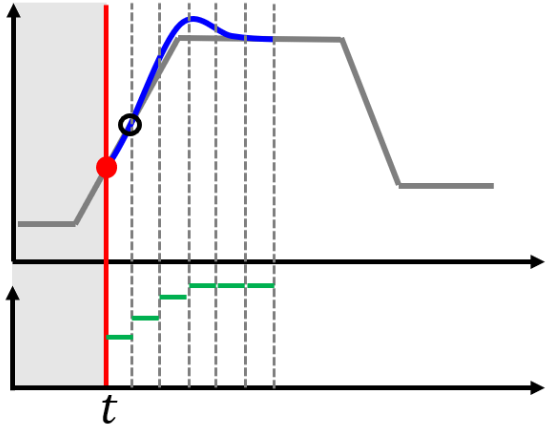
Model Predictive Control (MPC)

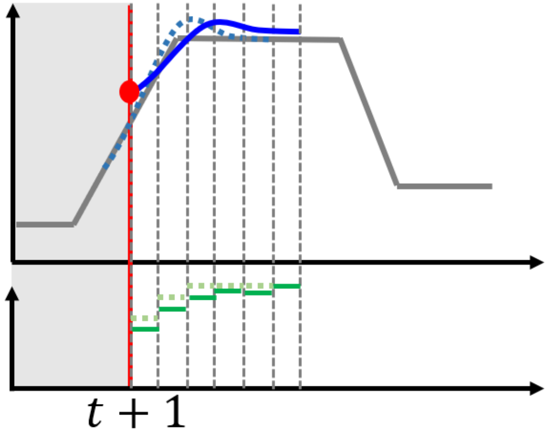
모델 (Model)을 가지고 미래를 예측 (Predictive) 을 하면서, 제어 (Control) 하겠다.

Step 1: 데이터를 활용하여 동적 모델 학습/업데이트

- 학습된 동적 모델: $G_{\tau+1}^{1:M} = f(G_\tau^{1:M}, Z_\tau^{1:N})$: f는 linear model, Neural Network, Graph Recurrent Network,....
- 학습된 관측 모델: $G_t^{1:M} = g(C_\tau^{1:N})$ (세팅 과정에서 글래스 TC 관측가능하다고 가정하므로 1차년도 사용X)

Step 2: MPC를 활용하여 실시간으로 최적 work set 값 최적화





$$\min_{Z_{t:t+W-1}^{1:40}} \frac{1}{W}\sum_{\tau=t}^{W}\sum_{i=1}^{140}\left(\hat{G}_{\tau+1}^{i} - R_{\tau+1}^{i}\right)^2$$

$\mathrm{s.t.}\ \hat{G}_{\tau+1}^{1:140} = f(G_\tau^{1:140}, Z_\tau^{1:40})$

Execute the first action: $Z_t^{1:40}$

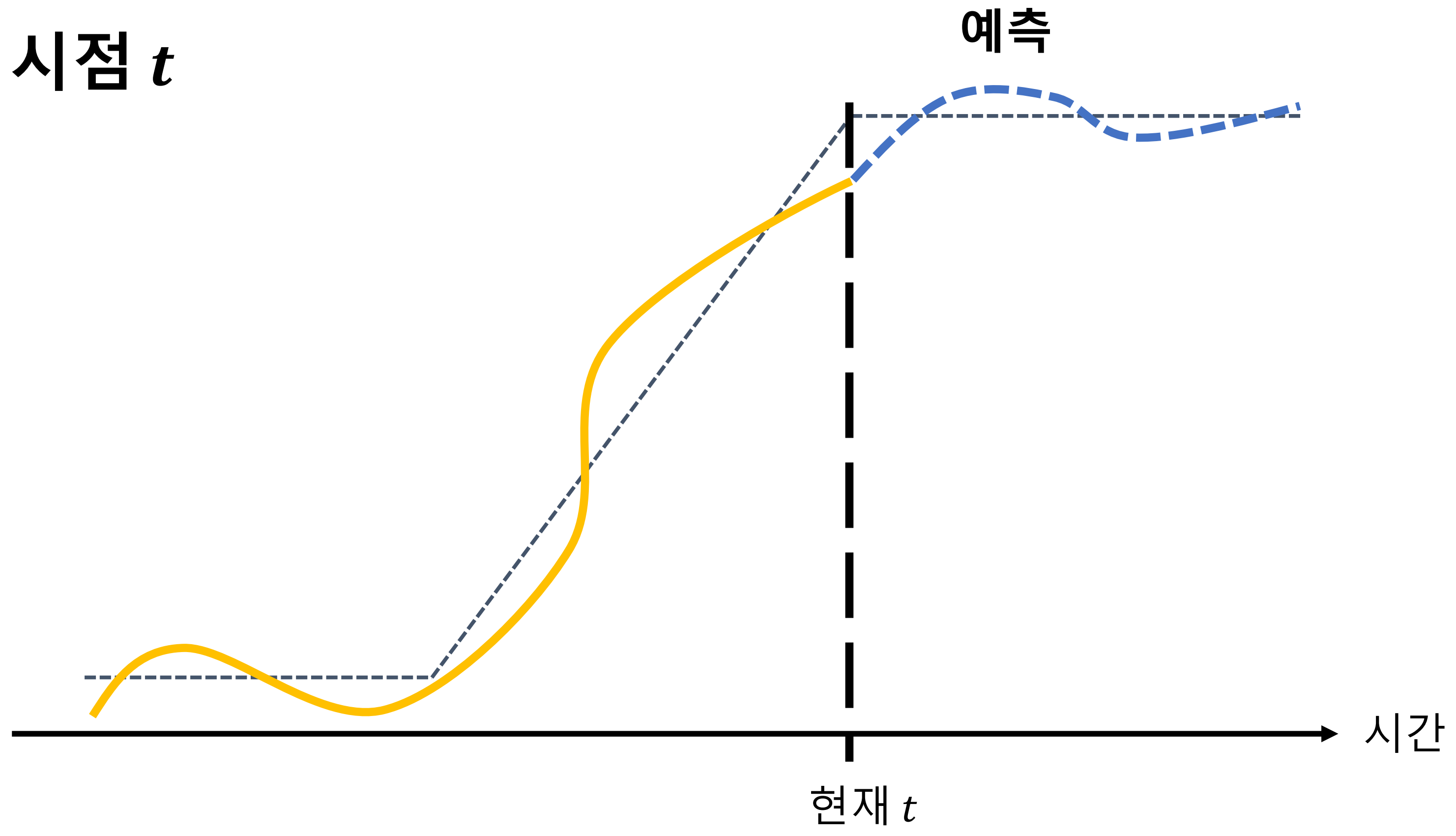
Observe $G_{t+1}^{1:140}$

$$\min_{Z_{t+1:t+W}^{1:40}} \frac{1}{W}\sum_{\tau=t+1}^{W+1}\sum_{i=1}^{140}\left(\hat{G}_{\tau+1}^{i} - R_{\tau+1}^{i}\right)^2$$

$\mathrm{s.t.}\ \hat{G}_{\tau+1}^{1:140} = f(G_\tau^{1:140}, Z_\tau^{1:40})$

...

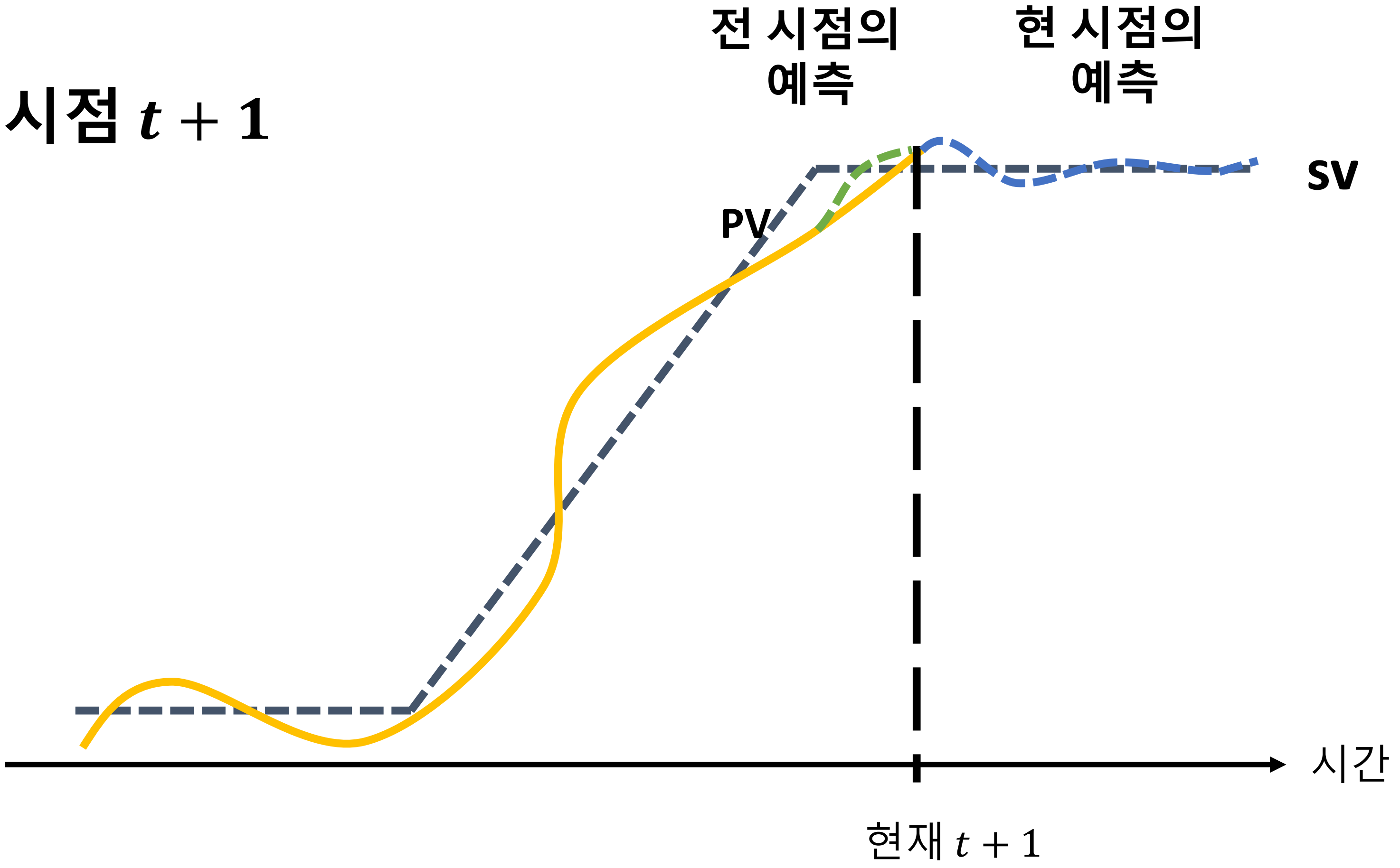
MPC를 활용한 제어 (1)



(1) $\{u_\tau\}_{\tau=t}^{t+H-1} \leftarrow \mathrm{MPC}\left(x_t, \{r_\tau\}_{\tau=t+1}^{t+H}; \hat{f}\right)$
(2) u_t 실행

MPC를 활용한 제어 (2)

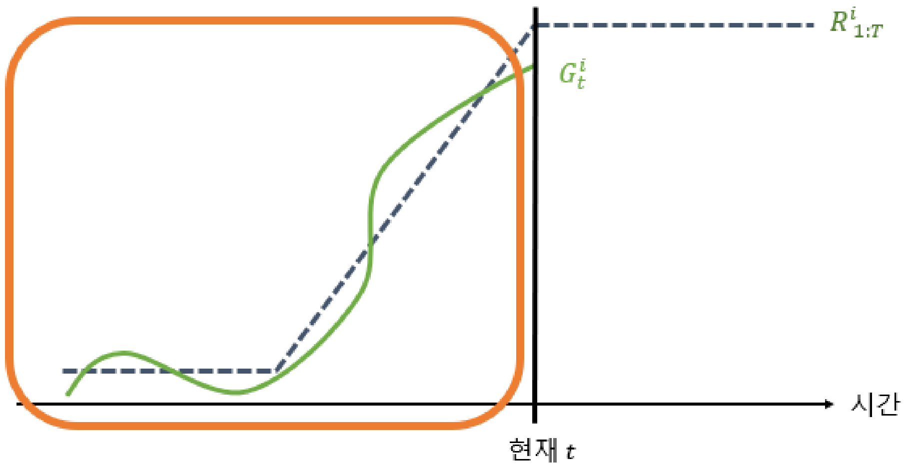
시점 $t + 1$

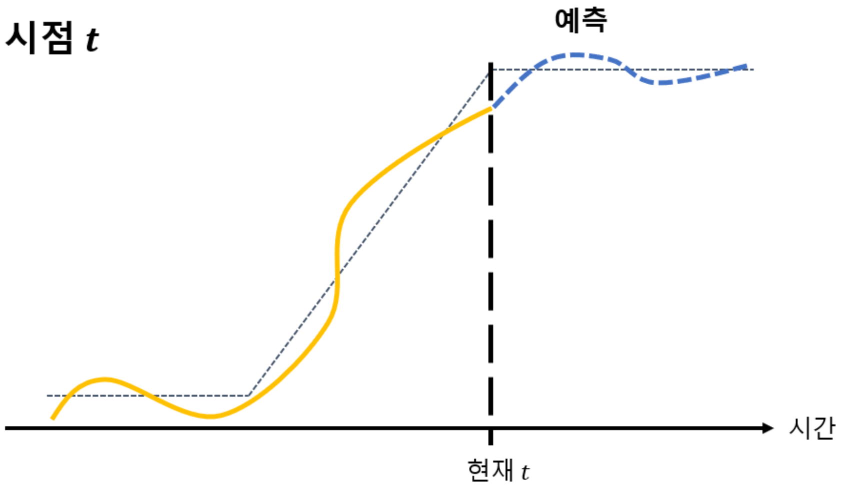


(1) $\{u_\tau\}_{\tau=t+1}^{(t+1)+H-1} \leftarrow MPC\left(x_{t+1}, \{r_\tau\}_{\tau=t+2}^{(t+1)+H}; \hat{f}\right)$

(2) u_{t+1} 실행

Comparison between PID and MPC





PID

Consider Past & Present error

MPC

Consider Past, Present & Future

Build Control Optimization Problem

- Control objective: Make glass TC track the reference trajectory
  - Learned model is used to predict the future glass TC
  - Valid workset value is between $\underline{u}$ and $\overline{u}$

- Mathematically,

$$
\begin{aligned}
&\min_{u_{0:H-1}} \frac{1}{H} \sum_{t=1}^{H} \|\hat{x}_t - r_t\|^2 \\
&s.t. \hat{x}_{t+1} = \hat{f}(\hat{x}_t, u_t; \theta) \\
&\underline{u} \le u_t \le \overline{u}
\end{aligned}
$$

- Wait, can we solve 'constrained optimization' via Gradient descent algorithm?

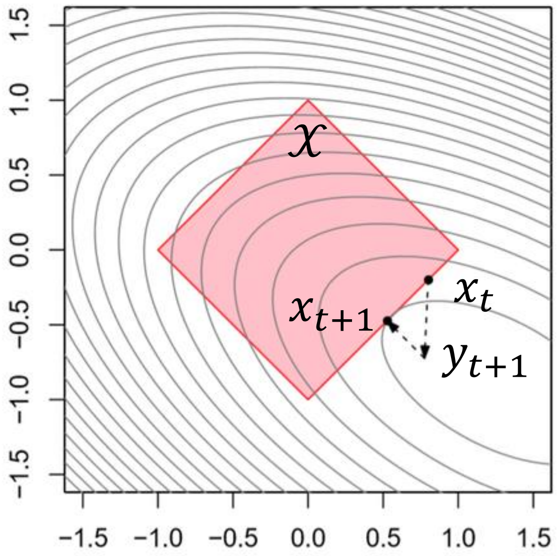
Projected Gradient Descent

- One way to solve constrained optimization

$$\min_{x \in \mathcal{X}} f(x)$$

- Key idea: After each gradient update, project update x into feasible set $\mathcal{X}$
  - $y_{t+1} = x_t - \alpha_k \nabla f(x_k), x_{t+1} = \mathrm{argmin}_{x \in \mathcal{X}} \|x - y_{t+1}\|^2$

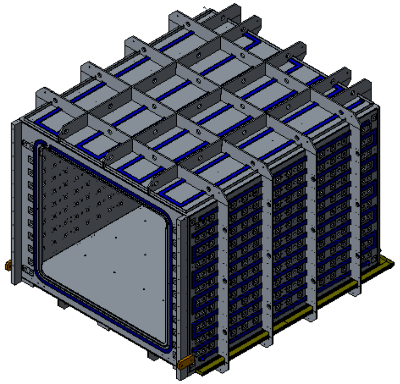
- If $f(\cdot)$ is convex and $\mathcal{X}$ is a convex set,
  - We can guarantee the global optimal solution
  - In our case, objective is quadratic & feasible space is box space: $[\underline{u}, \overline{u}]$

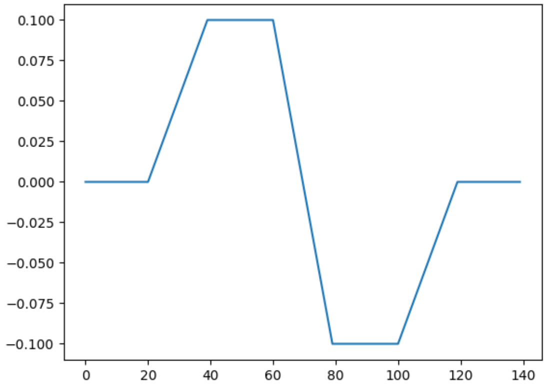


Code Exercise: Control Mini-Furnace via MPC + Multistep Linear

- Mini-Furnace
  - 5 Glass TC and 3 Workset
  - Goal: Control Workset so that glass TC tracks the reference trajectory

- Task 1. Train a Multistep linear model that mimics the Mini-furnace
- Task 2. Control the Mini-furnace via MPC with learned Multistep linear model





Reference Trajectory

Again, Real-World is Different…

- Need constraints for safety & stable control

- Limited time budget, need to boost the optimization speed

Boost Optimization Speed

- Good initial solution
  - Use optimal solution of previous timestep as an initial solution

- Learning rate scheduler
  - If the current solution is near-optimal, decrease the learning rate

- Time-out constraint
  - If computation time is over 5 seconds, finish algorithm and return the current-best solution

Safety Constraints

- In FTC, we cannot dramatically change Workset within 5 seconds

- Add smoothness term in the objective function

$$\min_{u_{0:H-1}} \frac{1}{H}\sum_{t=1}^{H}\|\hat{x}_t - r_t\|^2 + \lambda\frac{1}{H}\sum_{t=1}^{H}\|u_t - u_{t-1}\|^2$$
$$s.t.\ \hat{x}_{t+1} = \hat{f}(\hat{x}_t, u_t; \theta)$$
$$\underline{u} \le u_t \le \overline{u}$$

λ: smoothness lambda

Modify Objective Function for Better Control

- One of important control goal: reduce the overshoot on the stable range

- To reduce the overshoot, give different weights

$$\min_{u_{0:H-1}} \frac{1}{H}\sum_{t=1}^{H}\left(\omega\|\hat{x}_t - r_t\|_+{}^2 + \|\hat{x}_t - r_t\|_-{}^2\right) + \lambda\frac{1}{H}\sum_{t=1}^{H}\|u_t - u_{t-1}\|^2$$
$$s.t.\ \hat{x}_{t+1} = \hat{f}(\hat{x}_t, u_t; \theta)$$
$$\underline{u} \le u_t \le \overline{u}$$

ω: overshoot weight

Limit Feasible Space

- If reference temperature is $300℃$, then appropriate Workset will be between $300 \pm \alpha℃$

$$\min_{u_{0:H-1}} \frac{1}{H}\sum_{t=1}^{H}\left(\omega\|\hat{x}_t - r_t\|_+{}^2 + \|\hat{x}_t - r_t\|_-{}^2\right) + \lambda\frac{1}{H}\sum_{t=1}^{H}\|u_t - u_{t-1}\|^2$$

$$s.t.\, \hat{x}_{t+1} = \hat{f}(\hat{x}_t, u_t; \theta)$$

$$r_t - u_{range} \le u_t \le r_t + u_{range}$$

u_{range}: u_range

Review the Real MPC Code for FTC