Development of a Time-Series Surrogate Model for Predicting System Dynamics in KAIST-MMR under Load-following Operation

Transactions of Korean Nuclear Society Spring Meeting

Presenter : Jeong Yeol Baek
Advisor : Prof. Jeong Ik Lee

Nuclear Power & Propulsion Lab
Dept. of Nuclear & Quantum Engineering
Korea Advanced Institute of Science and Technology

KAIST

2 0 2 4 . 0 5 . 1 0

Contents

01. Introduction

- Background and motivation
- Research scope

02. Supervised-learning for Surrogate Model

- Data production
- Time-series surrogate model

03. Reinforcement-learning for Real-time Control

- Pre-training using surrogate model

04. Further Works

- Transfer learning using system analysis code
- Application to real world system

NPNP
Nuclear Power & Propulsion Laboratory

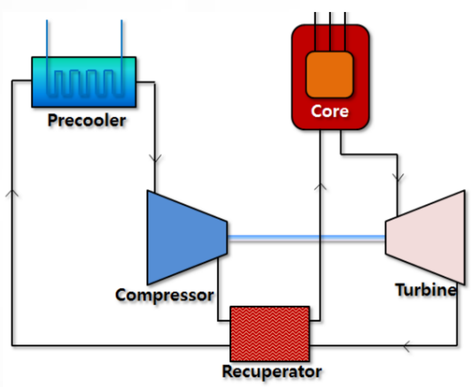
01
Introduction

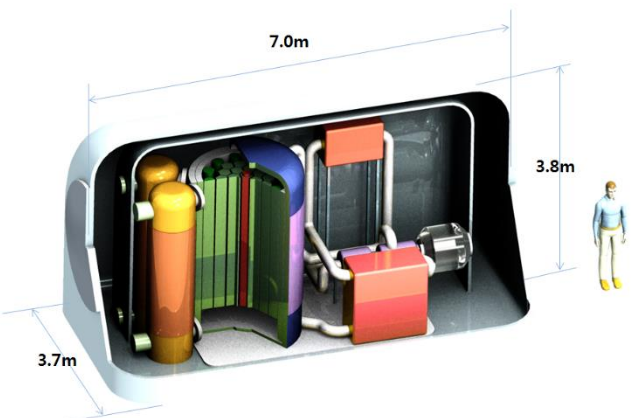
Introduction

Background and motivation

KAIST-MMR (Micro Modular Reactor)

- 10 MW_e class S-CO_2 direct cycle in combination with a fast neutron nuclear reactor
- Long life core (20 yrs) and enhanced safety feature
- Full modularization (reactor core, power conversion system)
- Transportable and installed to generate electricity in extreme environments (remote site)





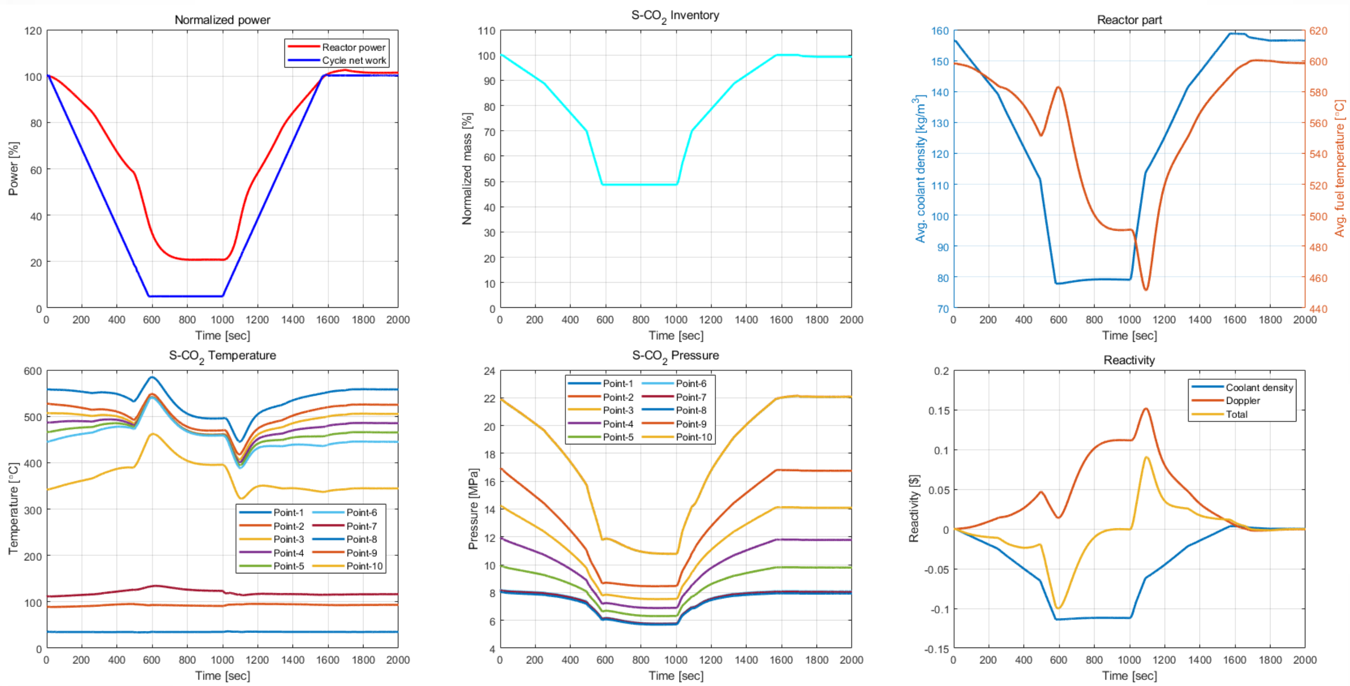
< Concept of KAIST-MMR >

Introduction

Background and motivation

KAIST-MMR (Micro Modular Reactor): load following capability

- Reactor power control & power conversion system control
- S-CO_2 coolant: compressible fluid and large density variations with pressure changes
- **Passive load following operation** using coolant inventory control



Introduction

Background and motivation

PID-based control of KAIST-MMR for load following operation

- Scenario (ramp rate & operation range) based **automatic** controller design under **limited optimization**
- Control objective
  - Regulation: Maintain a steady state value
  - Tracking: Follow a prescribed trajectory

  Stability, Performance optimization
- Control parameter
  - **Compressor inlet temperature**

$$E_{CIT} = \frac{CIT(t) - CIT_{design}}{CIT_{design}},\ d\dot{m}_c = k_p E_{CIT} + k_i \int_0^t E_{CIT}(t)\,dt + k_d \frac{dE_{CIT}}{dt}$$

  - **Turbine rotational speed**

$$\frac{d\omega}{dt} = \frac{(W_{turb} - W_{comp})\varepsilon_{generator} - W_{generator}}{\sum_i I_i \omega}$$

$$E_{TBP} = \frac{\omega(t) - \omega_{design}}{\omega_{design}},\ f_{TBP} = k_p E_{TBP} + k_i \int_0^t E_{TBP}(t)\,dt + k_d \frac{dE_{TBP}}{dt}$$

- Optimization parameter
  - **Inventory**

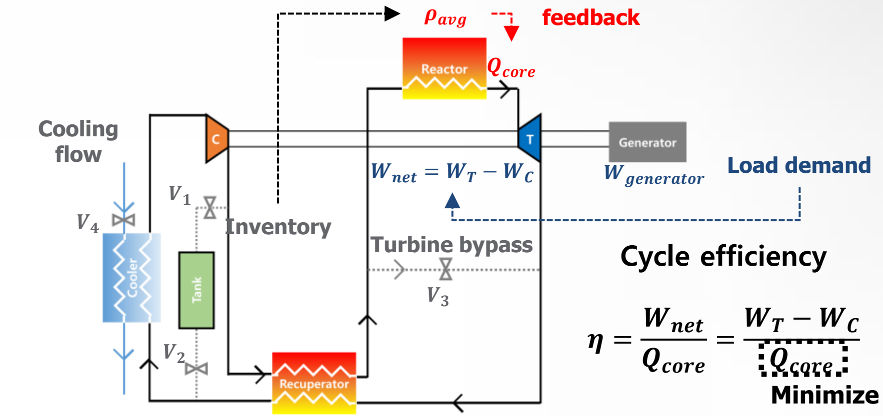
$$E_{IV} = \frac{M(t) - M_{target}}{M_{target}},$$

$$\dot{m}_{in} = k_p E_{IV} + k_i \int_0^t E_{IV}(t)\,dt + k_d \frac{dE_{IV}}{dt},$$

$$\dot{m}_{out} = k_p E_{IV} + k_i \int_0^t E_{IV}(t)\,dt + k_d \frac{dE_{IV}}{dt}$$

- Safety limit
  - Compressor surge margin

$$SM = \frac{\dot{m}_{comp} - \dot{m}_{surge}}{\dot{m}_{surge}}$$



Cycle efficiency

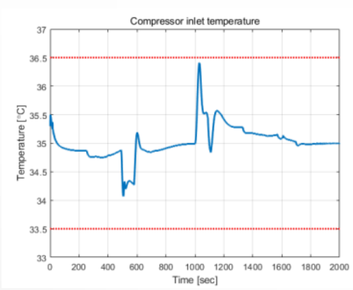
$$\eta = \frac{W_{net}}{Q_{core}} = \frac{W_T - W_C}{\boxed{Q_{core}}}$$

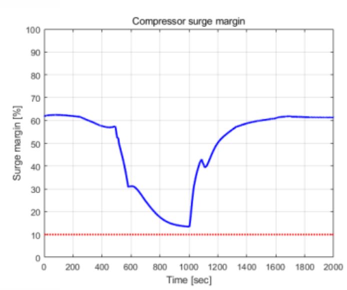
Minimize

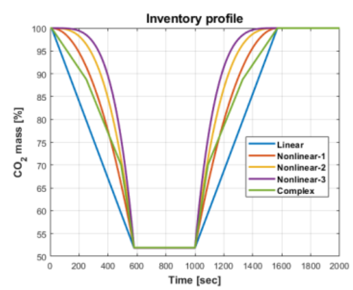
Dependent
- V_1: Control discharging flow → $\rho_{avg}\downarrow$ → $Q_{core}\downarrow$
- V_2: Control charging flow → $\rho_{avg}\uparrow$ → $Q_{core}\uparrow$
- V_3: Control turbine bypass flow → $W_T\downarrow$ (regulate ω)
- V_4: Control cooling flow → regulate CIT

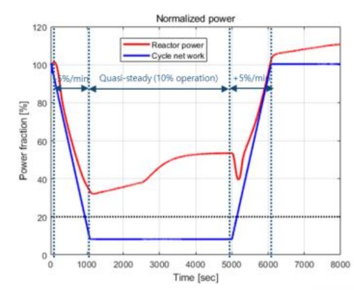
❖ **Optimum combination of $[V_1, V_2, V_3, V_4]$ with system stability**
  → Minimize Q_{core}

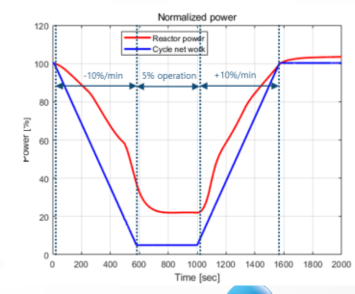
❖ **Many solutions satisfying a given scenario**







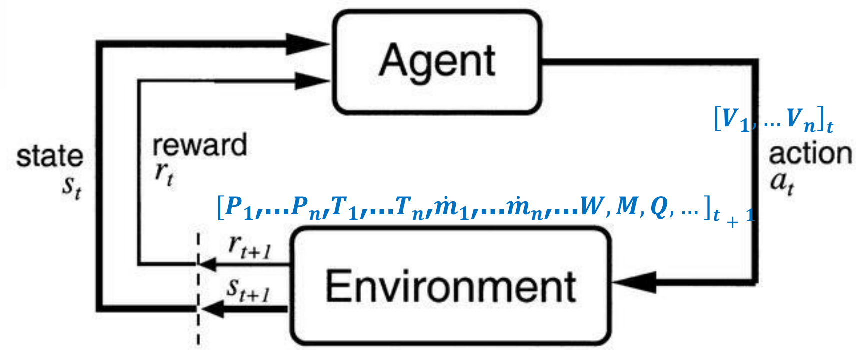




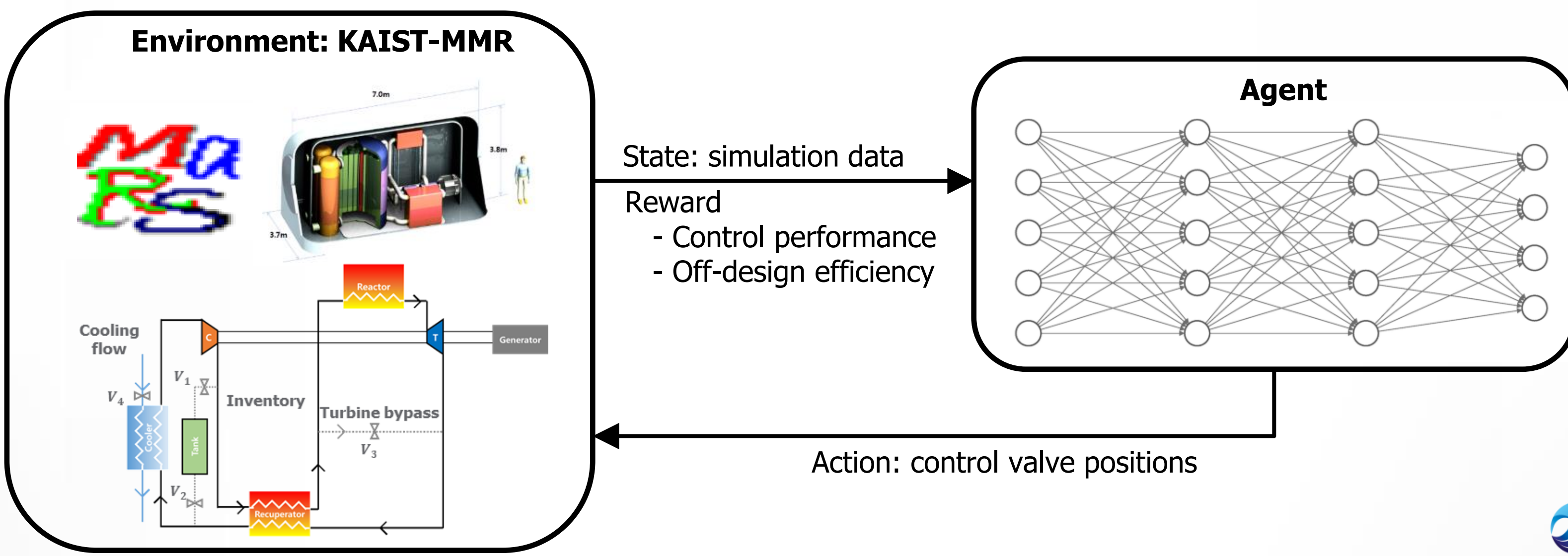
NPNP

Deep reinforcement learning (DRL) based control

- RL: Sequential decision making setup which consists of an agent interacting with an environment in discrete steps.
- RL problems are described as Markov Decision Processes (MDP)
  - State $s \in \boldsymbol{S}$
  - Action $a \in \boldsymbol{A}$
  - Reward $R_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
  - Policy $\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$
  - Discounting factor γ

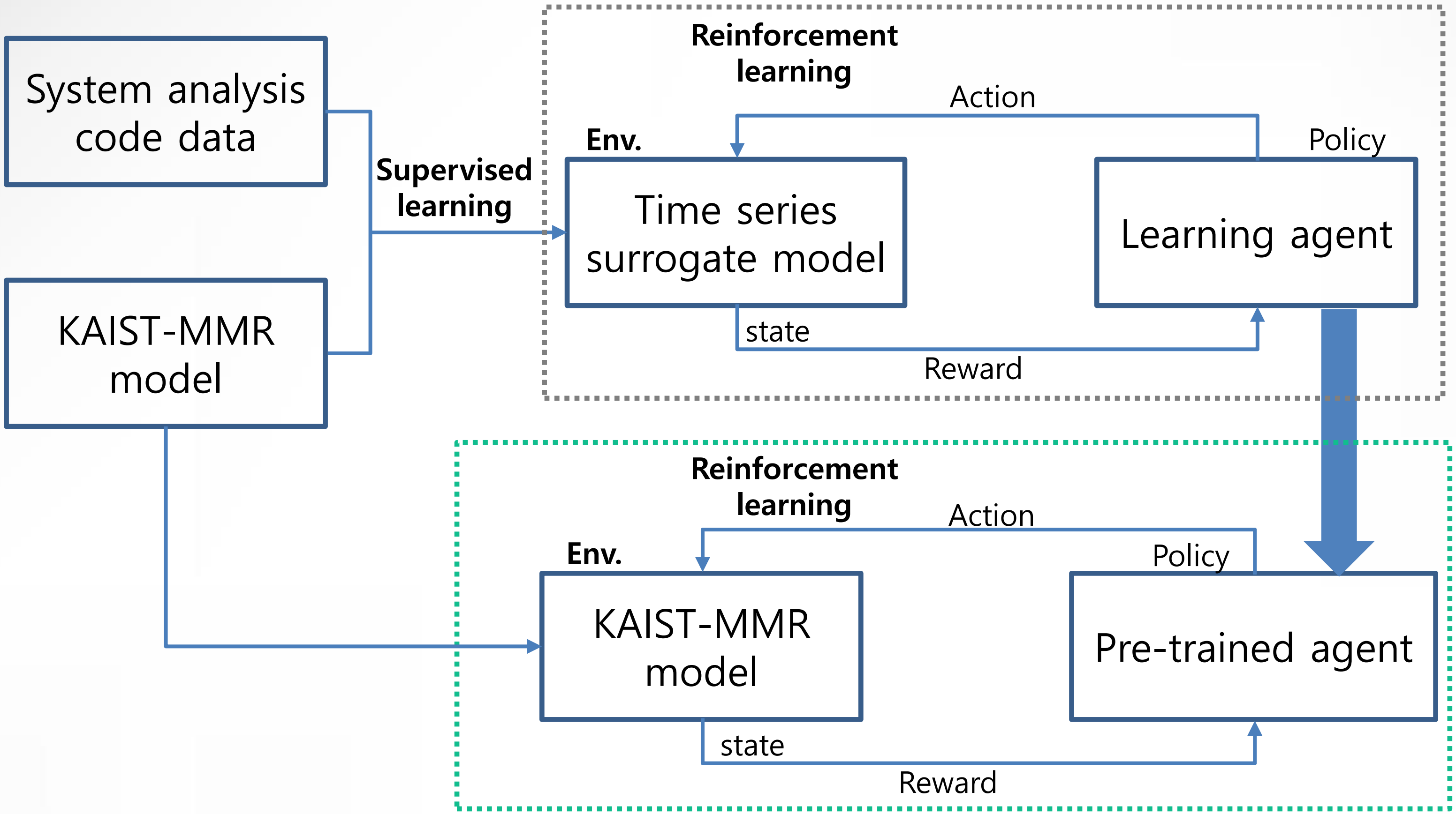


- It aims to increase the number of **independent state and control variables** as compared to the classic control environments



Introduction

Research scope



Pre-training

1. Training time
2. code stability

Transfer learning (Fine tuning)

02

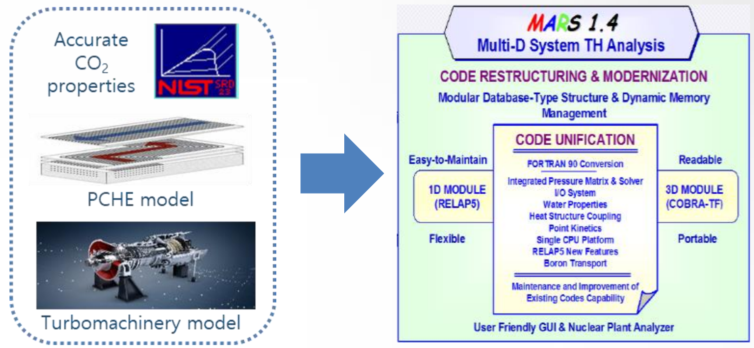
Supervised Learning

Supervised-learning

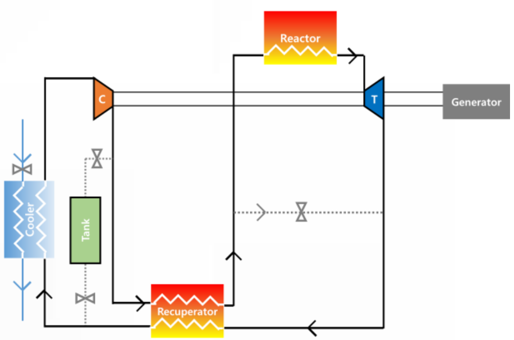
Data production

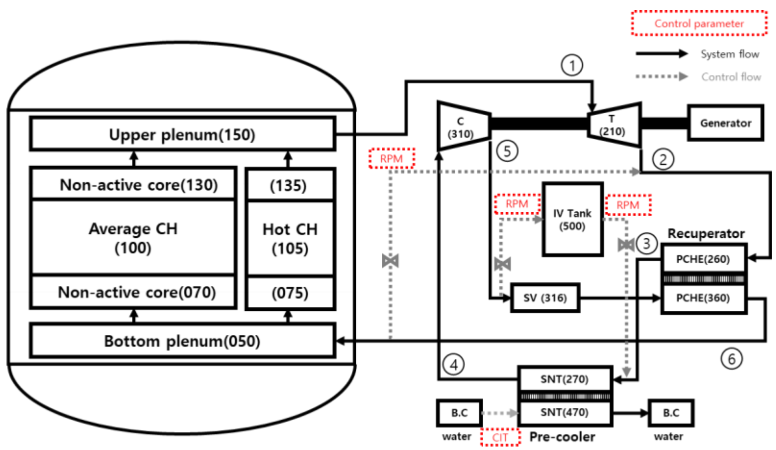
System modeling

- System analysis code: modified MARS code (MARS-CO_2)
- Based on MARS-KS 1.4 ver.
    1. NIST database for accurate calculation near the critical point
    2. PCHE heat transfer model for calculation (recuperator)
    3. Pham(CEA) turbomachinery similitude model



- Steady-state modeling including main components and control system



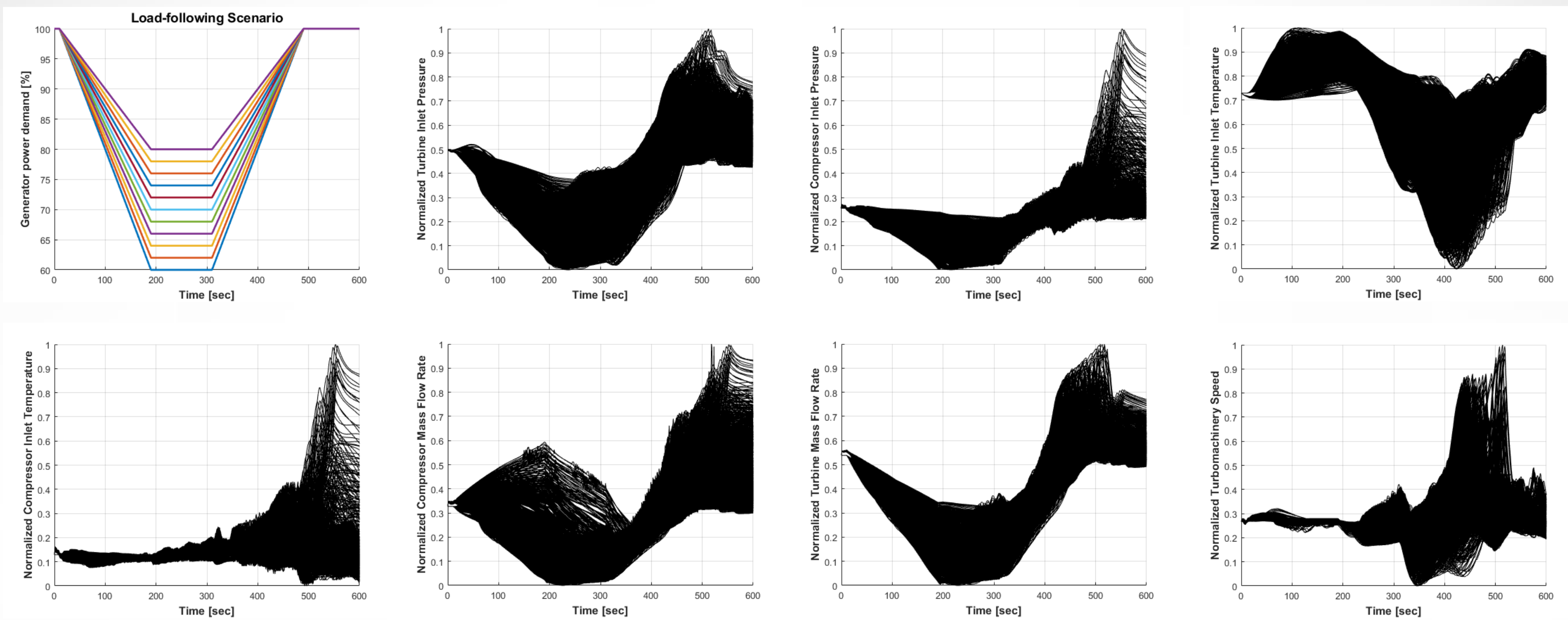


	$P\ [MPa]$	$T\ [K]$	$\dot{m}\ [kg/s]$
1	22.35	823.15	115.35
2	8.35	710.74	115.35
3	8.10	376.73	115.35
4	8.00	308.15	115.35
5	22.70	361.88	115.35
6	22.60	613.63	115.35

Supervised-learning

Data production

Load following simulation

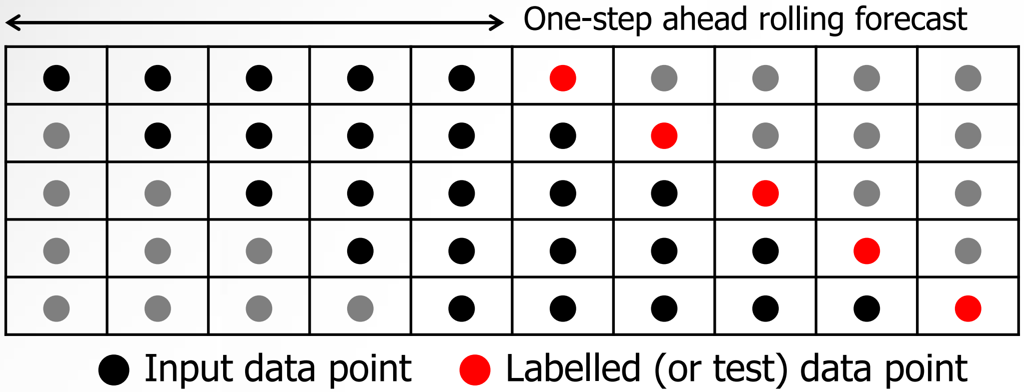


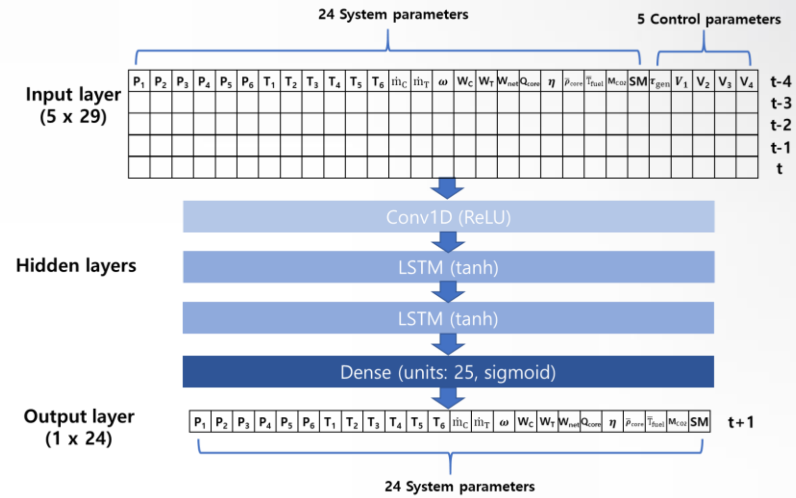
- Various PI gain values of control valves
- 9,900 simulation data with 5,949,900 data points

Supervised-learning

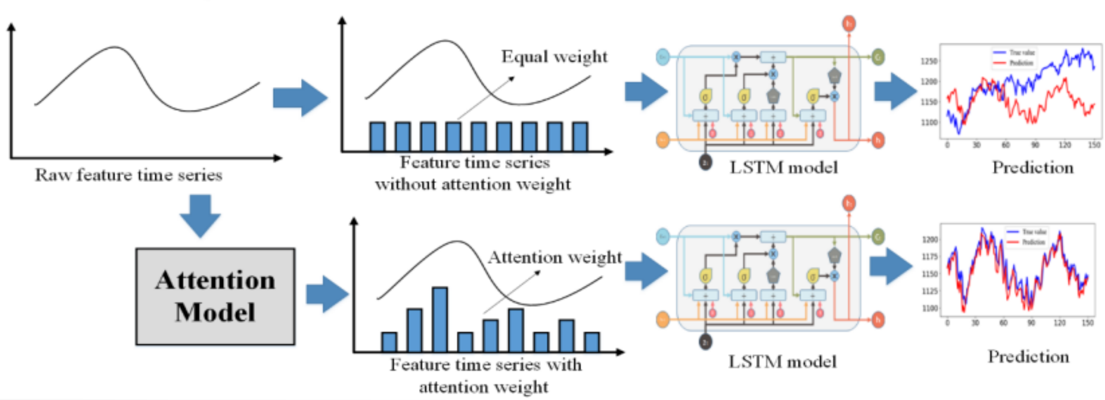
Time-series surrogate model

Rolling window forecast





- LSTM based multivariate time-series forecasting model
  - Conv1D-LSTM
  - Attention based LSTM



Training (80%) Validation (10%) Test (10%)

Number of nodes	MAE	
	Conv1D-LSTM	Attention based LSTM
32	3.0e-3	6.37e-4
64	1.9e-3	5.17e-4
128	1.8e-3	4.58e-4
256	1.7e-3	4.41e-4
512	1.6e-3	4.36e-4
1024	1.5e-3	4.17e-4

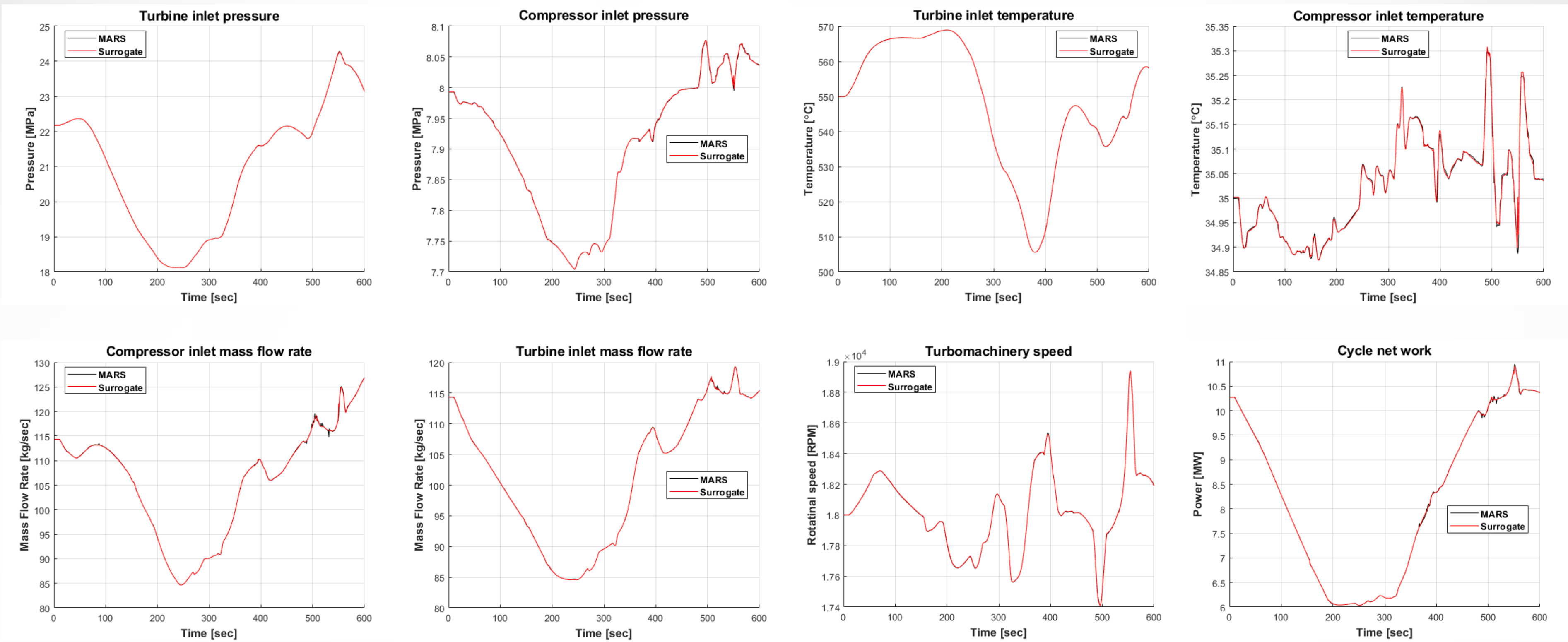
$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_{pred,i} - y_{MARS,i}\right|$$

NPNP

Supervised-learning

Time-series surrogate model

Surrogate model test results



< Comparison of MARS code data and predicted values from the surrogate model >

- Test results show very good agreement with MARS simulation data

03

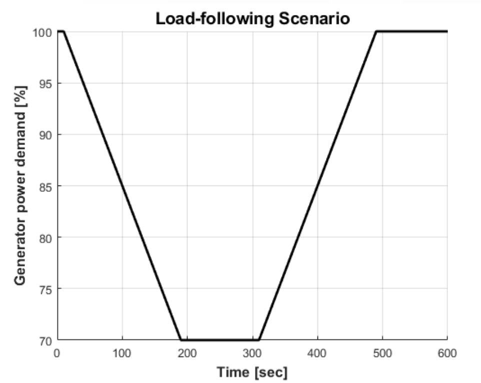
Reinforcement Learning

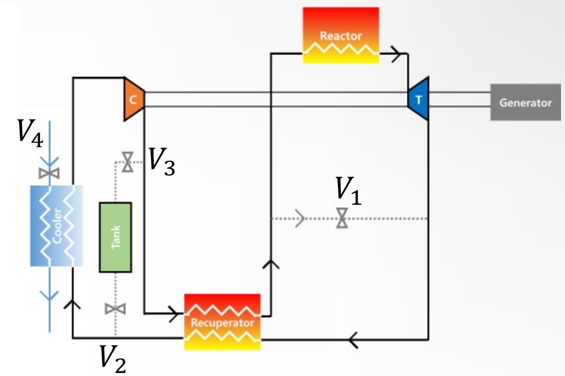
Reinforcement-learning

Pre-training using surrogate model

RL environment and training conditions

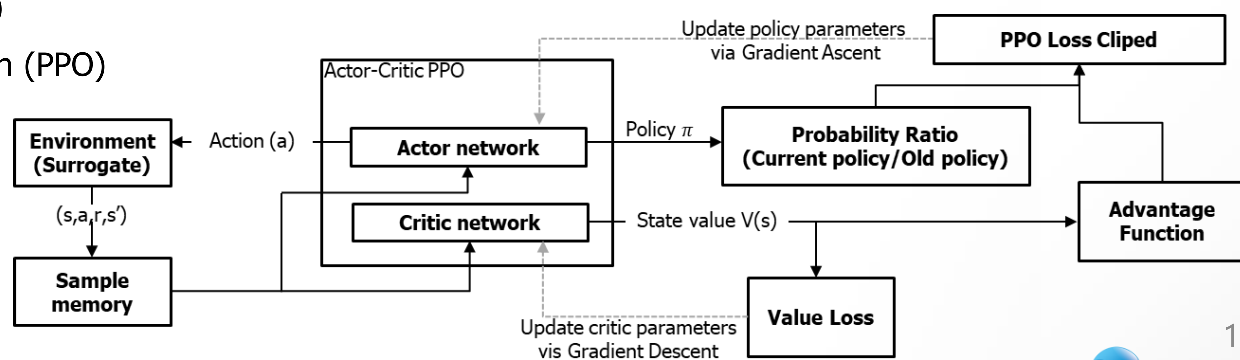
- Scenario: 100% - 70% - 100% operation (10%/min)
- Action space: Box(-1, 1, (3,)) control valves
  1. TBP (-1: V_1 fully closed, 1: V_1 fully opened)
  2. Inventory (0: V_2 & V_3 closed,
     -1: V_2 fully opened, 1: V_3 fully opened)
  3. CIT (-1: V_4 fully closed, 1: V_4 fully opened)





- Observation: Box(0, 1, (24,)) system parameters
- Rewards
  1. RPM
  2. Efficiency
  3. CIT
  4. Alive: +1
  5. Abnormal termination (penalty)

- Algorithm: Proximal policy optimization (PPO)
- Max episode length: 600
- Max training length: 3e7

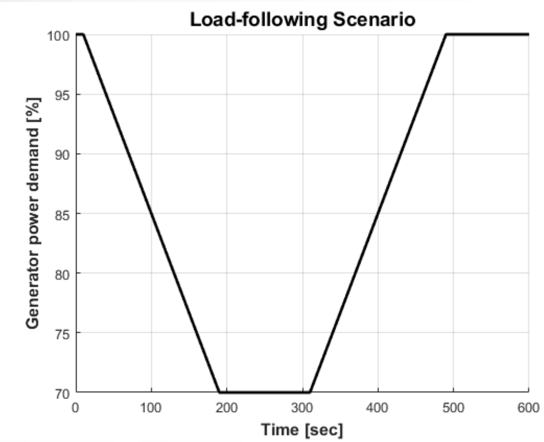


NPNP

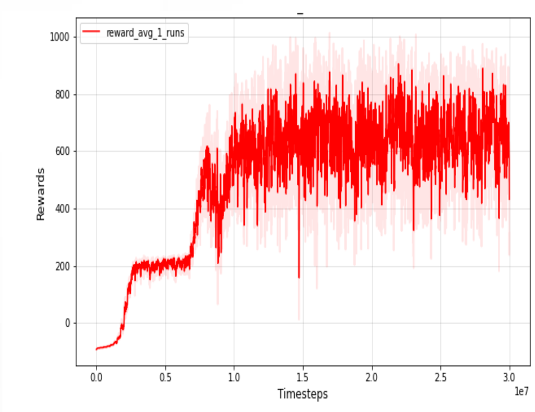
Reinforcement-learning

Pre-training using surrogate model

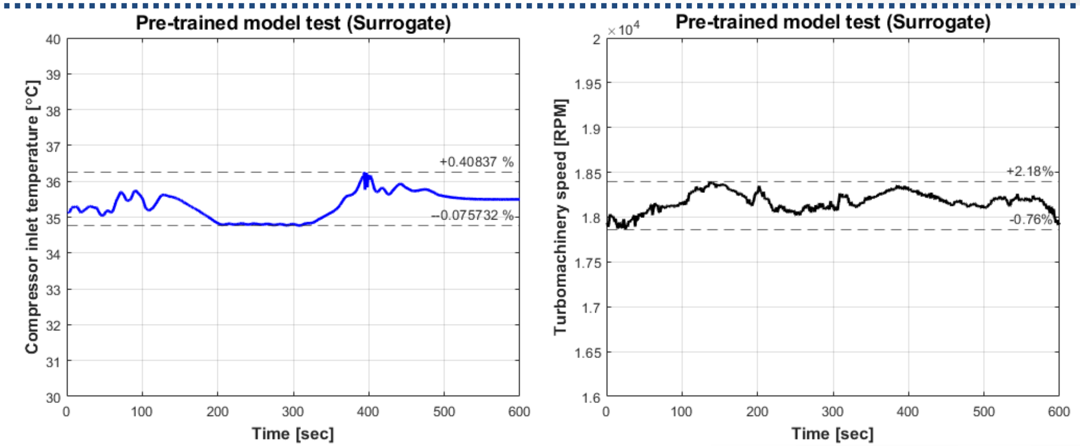
Pre-training results and control test



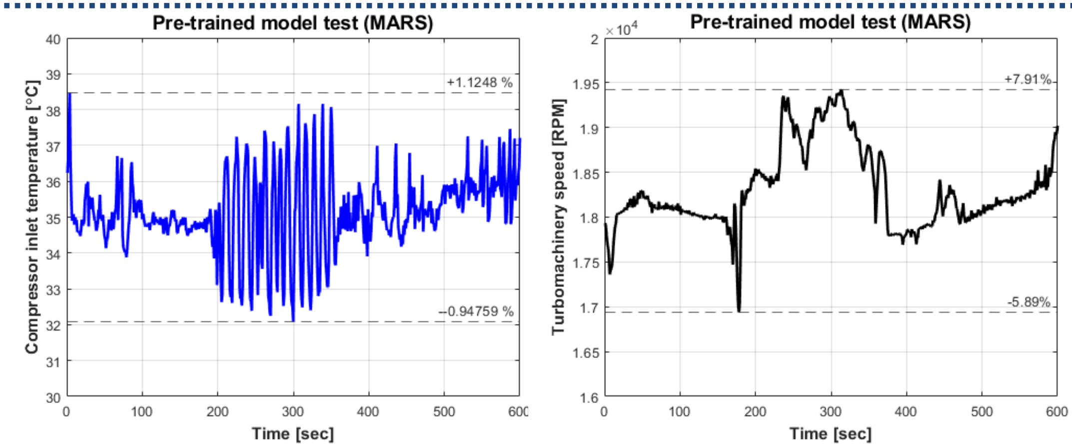
< Training & test scenario >



< Pre-training result >



< Control test results in surrogate environment (CIT & RPM) >



< Control test results in MARS environment (CIT & RPM) >

- Pre-trained agent shows reasonably good control performance under the trained scenario.
- It also performs control in the correct direction in MARS environment, but needs to be improved further with post-learning.

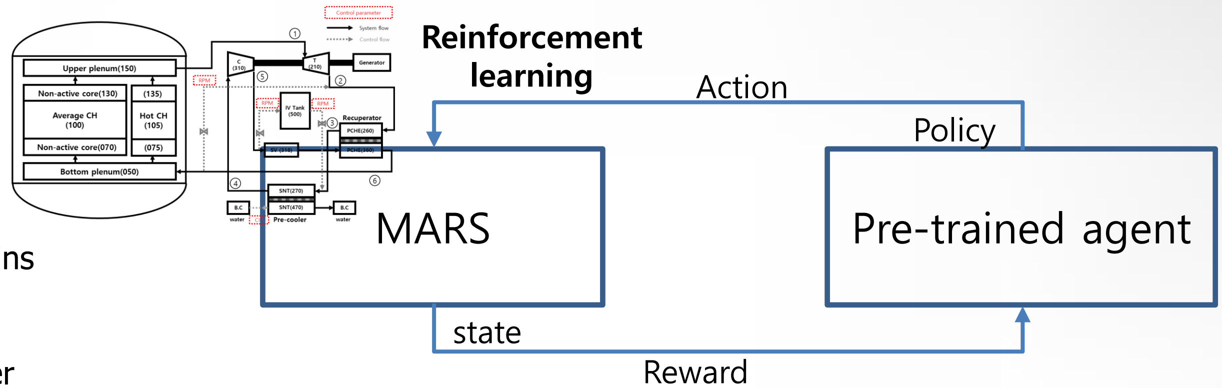
04
Further Works

Further Works

Transfer learning using system analysis code

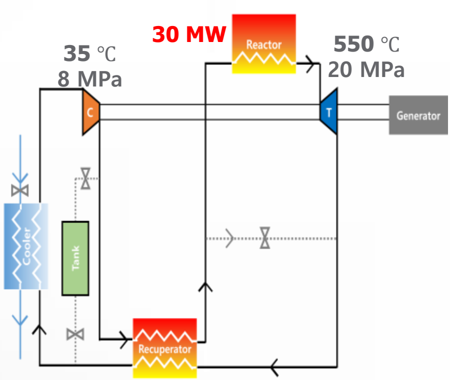
- Fine tuning (lower training time steps)
- Optimizer: Adam → SGD
- Learning rate ↓
- Training time comparison
  - Surrogate model: 3e7 steps ~570 mins
  - MARS: 1e5 steps ~2280 mins

  Surrogate model → ~1200 times faster

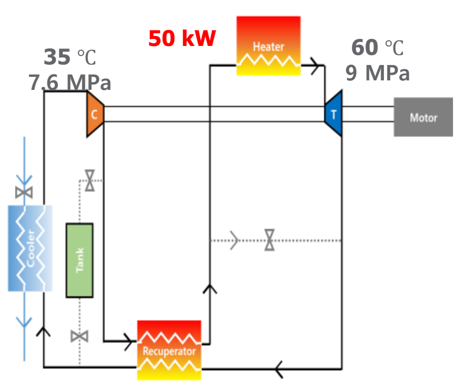


Application to real world system

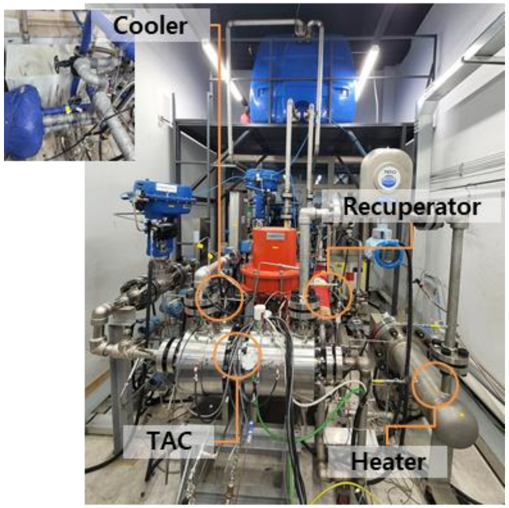
- Hardware validation of DRL-based control methodology



< Layout of KAIST-MMR >



< Layout of ABC test loop >



< ABC test loop >

< Inventory tank >

NPNP

Thank you for your attention

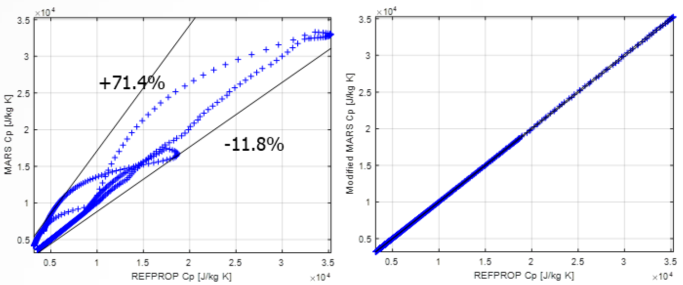
Q & A

Appendix

System analysis code: MARS code modification

1. Physical properties of CO_2

- NIST database for accurate calculation near the critical point



2. PCHE heat transfer model

- For design and off-design calculation of recuperator

$$Nu = 0.0292Re^{0.8137}$$

3. Turbomachinery model

- CEA similitude model & performance maps

CEA Similitude Model

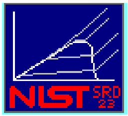
Inlet Conditions: $T, P, \dot{m}, RPM$ → Performance: Pr, eff

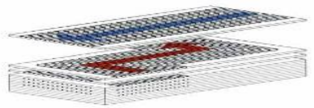
Corrected properties

$$\dot{m}_c = \left(\frac{\dot{m}\sqrt{n_sRZT}}{n_sP}\right)\left(\frac{n_sP}{\sqrt{n_sRZT}}\right)_{Ref} \quad \Delta H_c = \left(\frac{\Delta H}{n_sRZT}\right)(n_sRZT)_{Ref}$$

$$N_{rpm,c} = \left(\frac{N_{rpm}}{\sqrt{n_sRZT}}\right)\left(\sqrt{n_sRZT}\right)_{Ref} \quad n_s = -\frac{v}{P}\left(\frac{\partial P}{\partial v}\right)_s$$

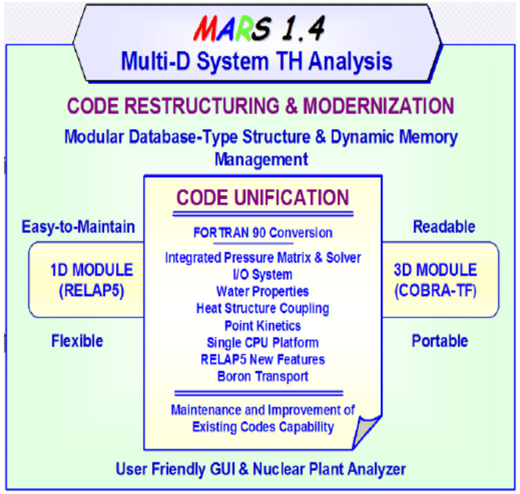
Accurate CO_2 properties



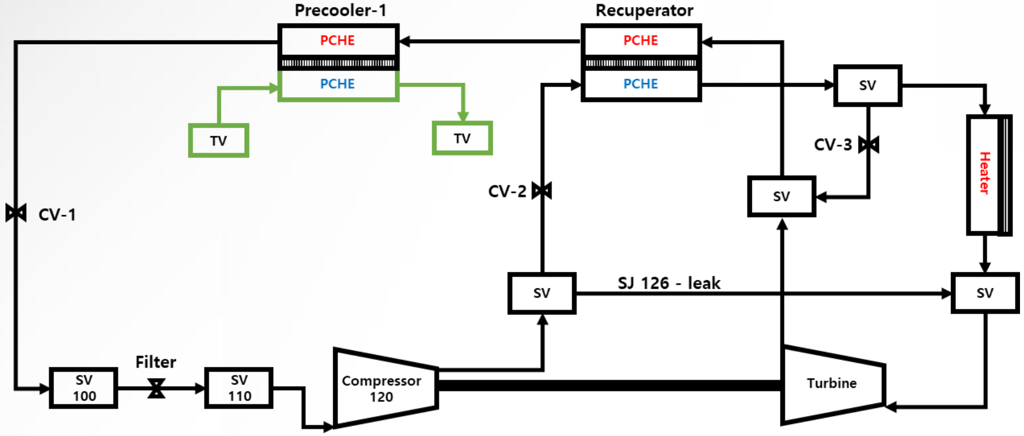


PCHE model

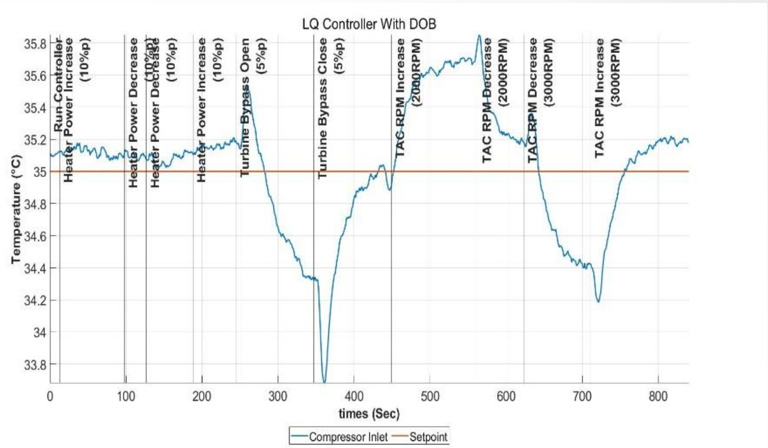
Turbomachinery model



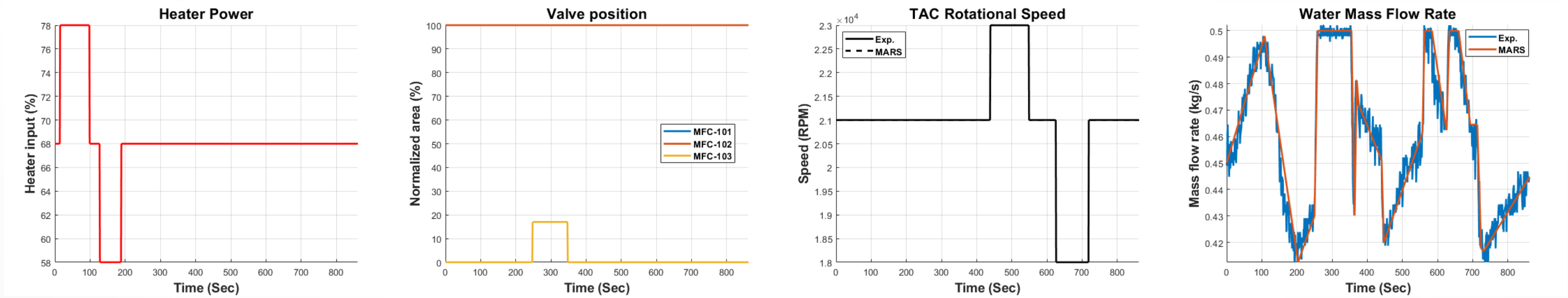
System analysis code: loop modeling & code validation



< Schematic of ABC loop modeling in MARS code >

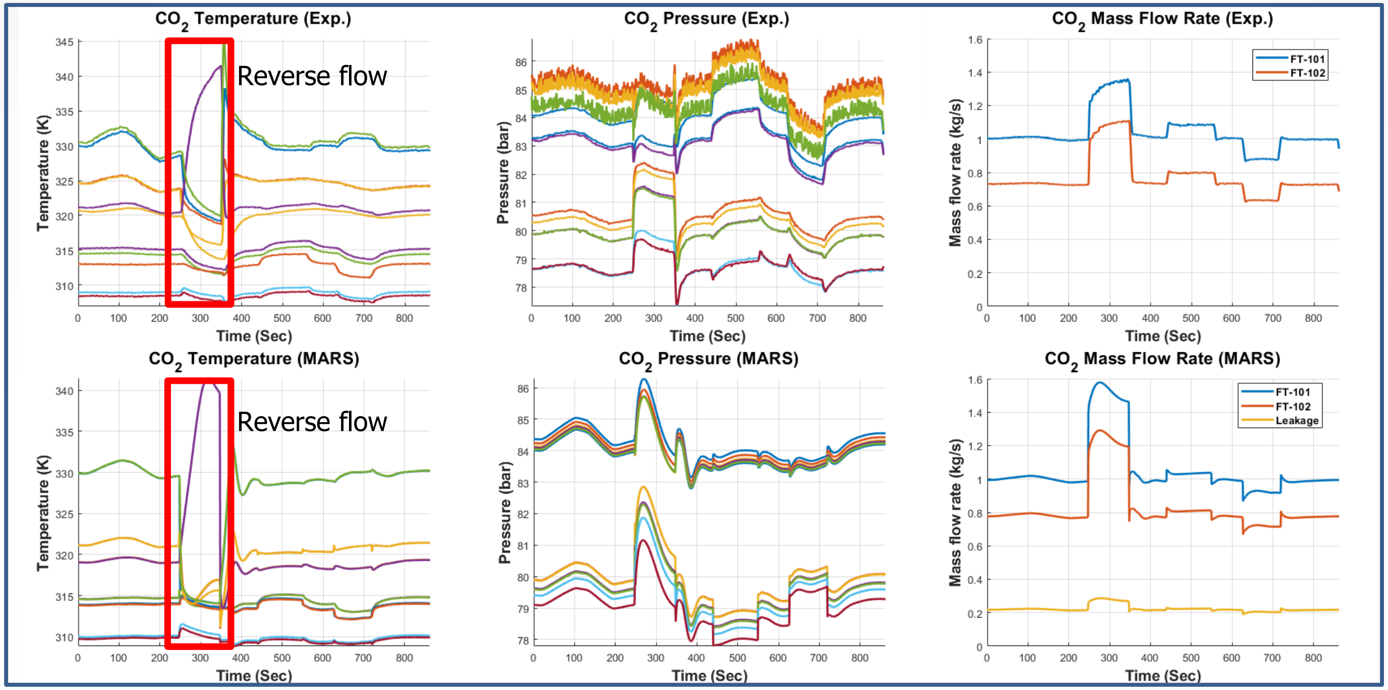


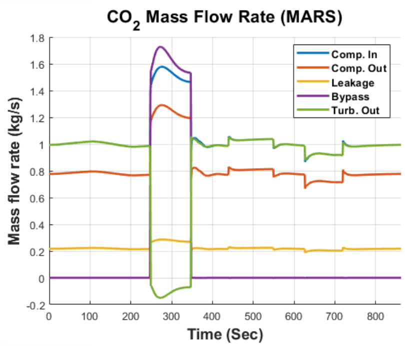
< Previous CIT control test data >

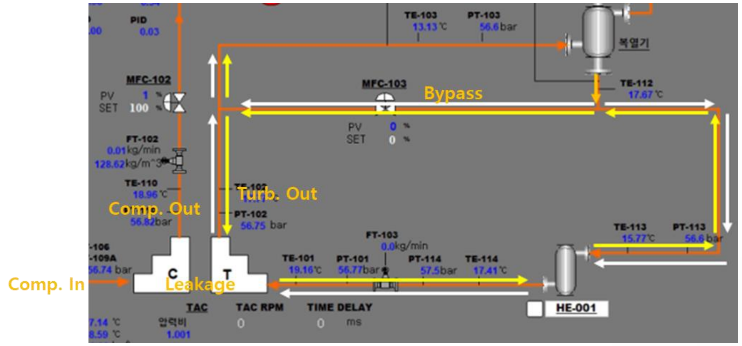


< Transient scenario in MARS code simulation >

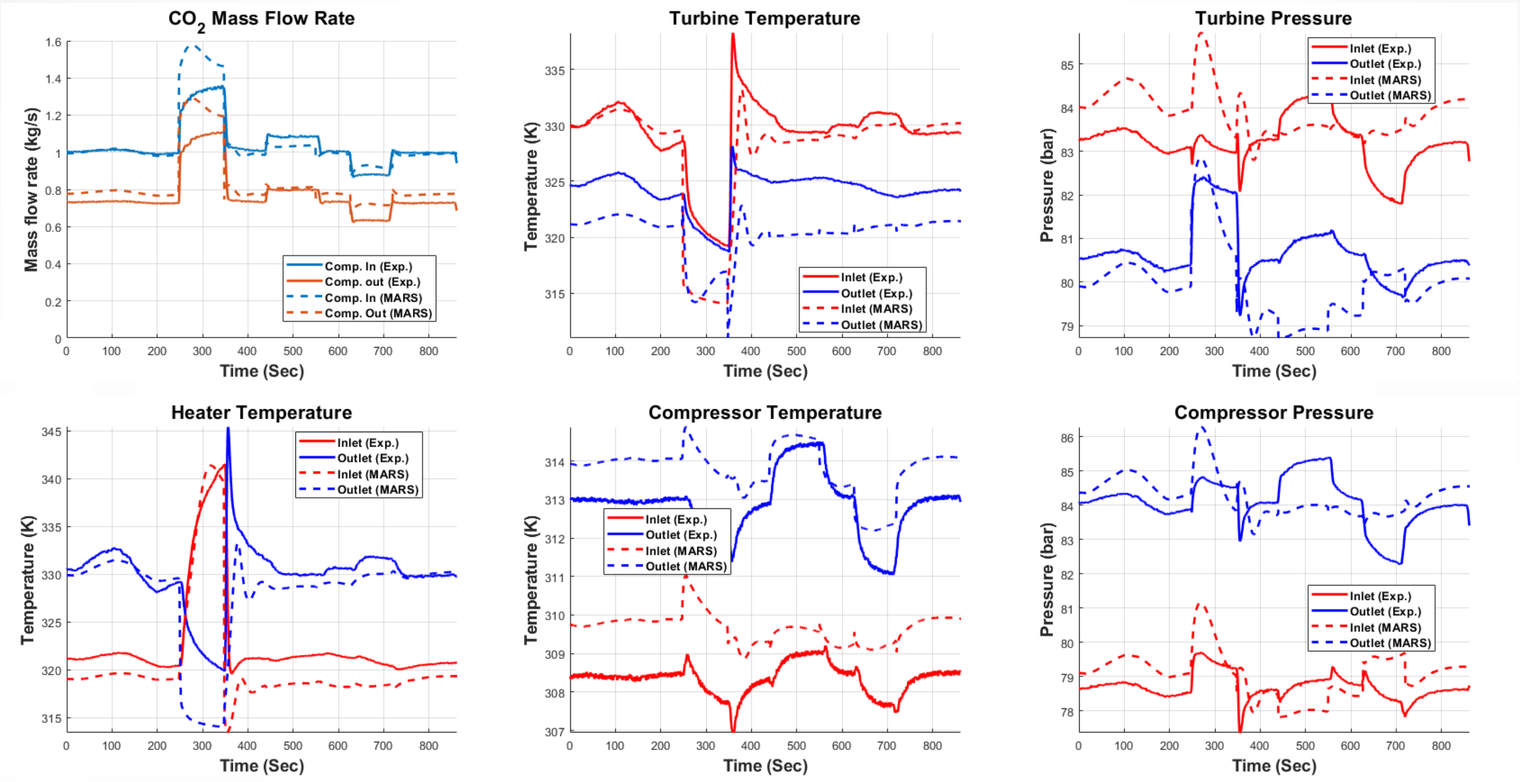
System analysis code: loop modeling & code validation





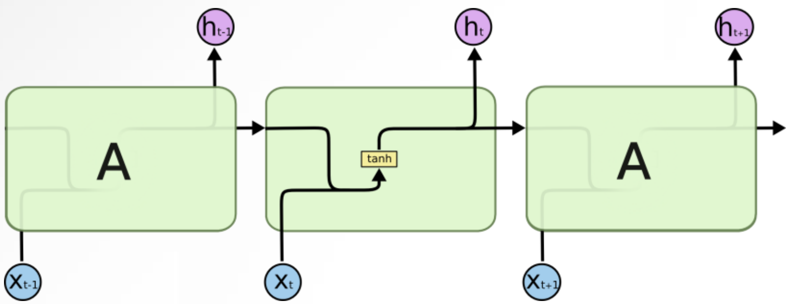


System analysis code: loop modeling & code validation

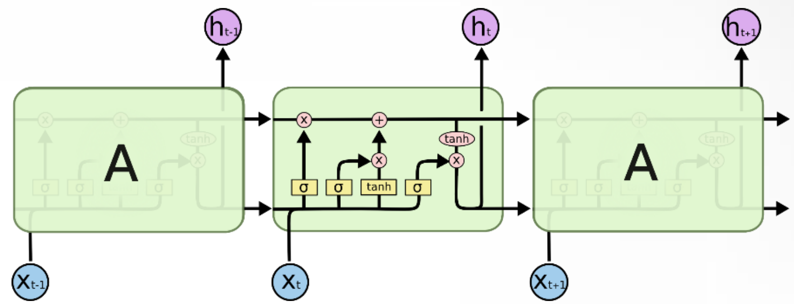


- MARS code can well simulate the dynamic characteristics of the ABC test loop
- It is possible to predict not only the **normal** state but also the **abnormal** state of the system.

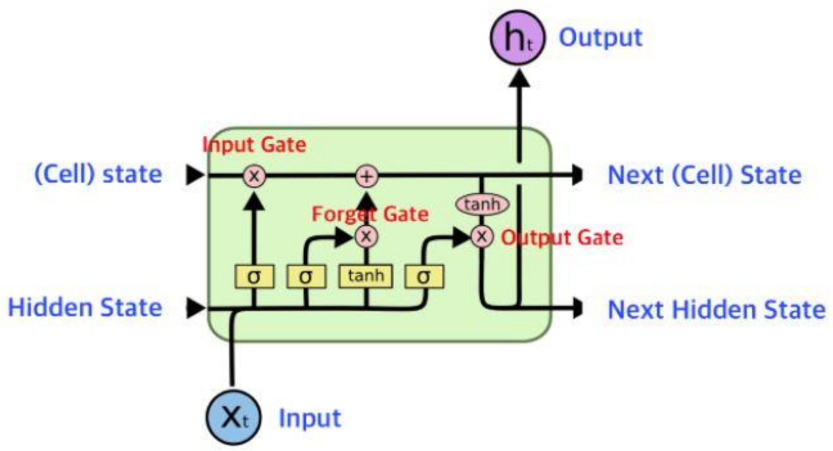
Long Short Term Memory (LSTM) model



< Recurrent neural network (RNN) >



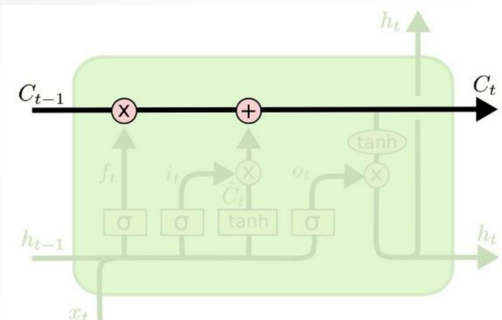
< LSTM >



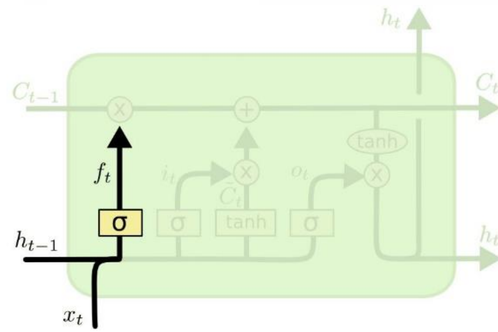
- Model to solve the long-term dependencies in traditional RNNs.
- To predict future data by considering not only the previous data, but also the past data more macroscopically
- 6 parameters & 4 gates

Long Short Term Memory (LSTM) model

Cell state

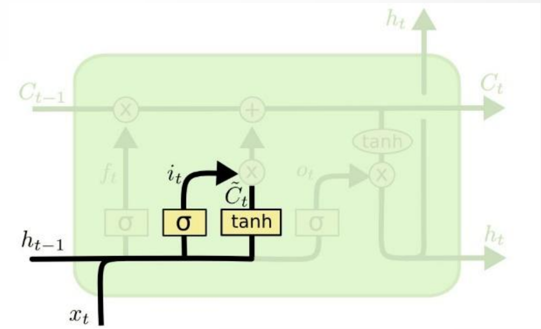


Forget gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \;+\; b_f\right)$$

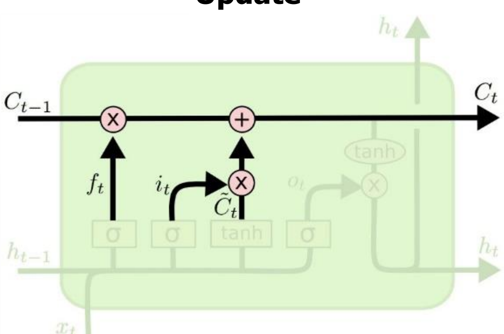
Input gate



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \;+\; b_i\right)$$

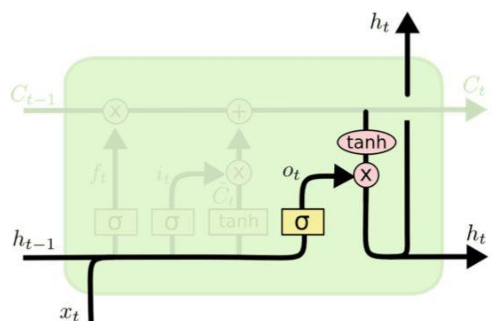
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \;+\; b_C)$$

Update



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output gate

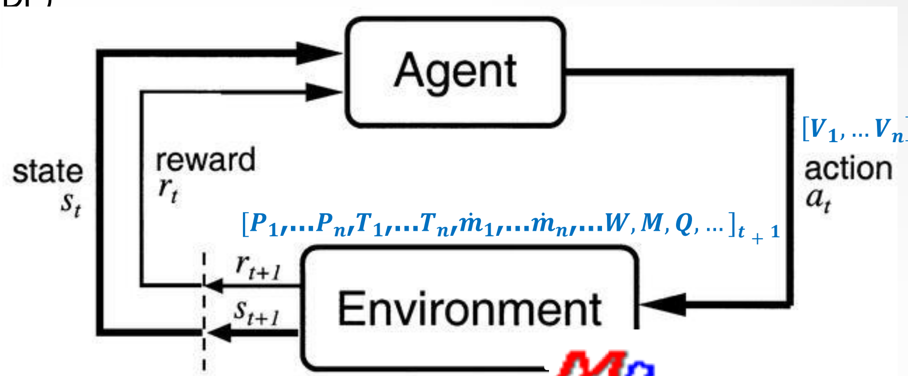


$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \;+\; b_o\right)$$

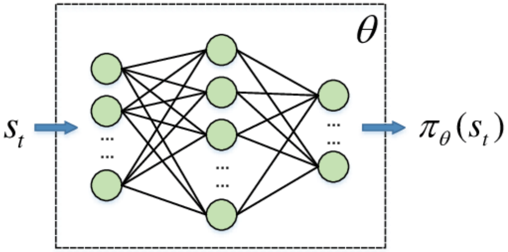
$$h_t = o_t * \tanh\left(C_t\right)$$

Deep Reinforcement learning

- RL: Sequential decision making setup which consists of an agent interacting with an environment in discrete steps.
- RL problems are described as Markov Decision Processes (MDP)
  - State $s \in \boldsymbol{S}$
  - Action $a \in \boldsymbol{A}$
  - Reward $R_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
  - Policy $\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$
  - Discounting factor γ



- Objective: maximize the expected return by choosing an optimal policy
  - Return $G_t = \sum_0^\infty \gamma^k R_{t+k+1}$
  - State-value function $V_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$,$V_\pi^*(s) = \max_\pi V_\pi(s)$
  - Action-value function $Q_\pi(s,a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a]$, $Q_\pi^*(s,a) = \max_\pi Q_\pi(s,a)$



- Value based RL (DQN)
  - Minimize loss function $L(\theta)$
  - $\theta_{k+1} \leftarrow \theta_k - \alpha \cdot \nabla_\theta L(\theta)$ (gradient descent)
  - $\pi^*(a|s) = \begin{cases} 1 & if\ a = argmax_a\ Q_\pi^*(s,a) \\ 0 & otherwise \end{cases}$

- Policy based RL
  - Maximize objective function $J(\theta) = V_{\pi_\theta}(S_0)$
  - $\theta_{k+1} \leftarrow \theta_k + \alpha \cdot \nabla_\theta J(\theta)$ (gradient ascent)
  - Action choice: *softmax*

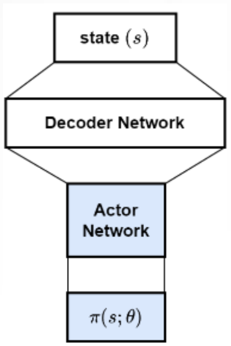
RL algorithm: Proximal Policy Optimization (PPO)

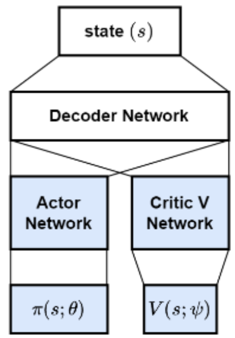
- Policy gradient (PG): $\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) \cdot Q_{\pi_\theta}(s,a)]$

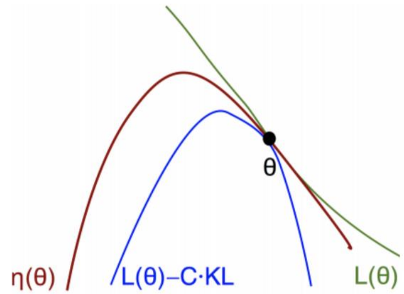
$\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) \cdot G_t]$

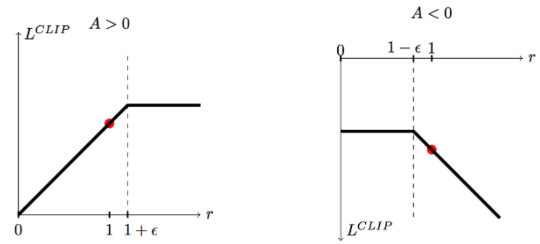
$\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) \cdot A_W(s,a)]$

PG/REINFORCE	Actor Critic (A2C)	Trust Region Policy Optimization (TRPO), 2015	Proximal Policy Optimization (PPO), 2017

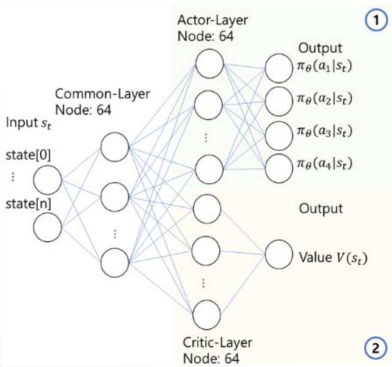








$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}$$



Advantage function

$A_{\pi_\theta} = Q_{\pi_\theta}(s,a) - V_{\pi_\theta}(s)$

$J(\theta) - J(\theta_{old}) = L(\theta) > 0$

Kullback-Leibler (KL) divergence

$D_{KL}(\pi_{\theta_{old}}(\cdot|s) \parallel \pi_\theta(\cdot|s)) \le \delta$

Clip parameter

$1-\epsilon \le r(\theta) \le 1+\epsilon$