OPERATION ISSUES ON CARBON CAPTURE PROCESSES OF POWER PLANTS

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2019

Ziang Li
School of Electrical and Electronic Engineering

To My Parents

Contents

List of Tables

List of Figures

Abbreviations

AIC Akaike's information criterion

AIC$_c$ Akaike's information criterion with a correction for finite sample sizes

BIBO bounded-input-bounded-output

BIC Bayesian information criterion

CCS carbon capture and storage

CFDL compact form dynamic linearisation

ETS emission trading system

GPC generalised predictive control

ICE Intercontinental Exchange

KL Kullback-Leibler

LHV lower heat value

MDP Markov decision process

MEA mono-ethanol-amine

MFAC model-free adaptive control

MIMO multi-input-multi-output

MLE maximum likelihood estimate

MPC model predictive control

OM operation and maintenance

PCC post-combustion carbon capture

PDF probability density function

PFDL partial form dynamic linearisation

PI proportional-integral

PID proportional-integral-derivative

PPD pseudo-partial derivative

RGA relative gain array

SISO single-input-single-output

TD temporal difference

w.r.t. with respect to

Notations

1. Mathematics

$\mathbf{A}(q) = 1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_{n_a} q^{-n_a}$

$\mathcal{A}_i$ set of actions for agent i

$\mathcal{A}$ set of actions for an agent

$\mathbf{b}^l$ bias vector of layer l for neural network

$\bar{b}_1$ tuning parameter of MFAC

$\bar{b}_2$ tuning parameter of MFAC

$\mathbf{B}(q) = b_0 + b_1 q^{-1} + b_2 q^{-2} + \cdots + b_{n_b-1} q^{-n_b+1}$

$\tilde{C}$ cost function for neural network

$\mathbf{C}(q) = 1 + c_1 q^{-1} + c_2 q^{-2} + \cdots + c_{n_c-1} q^{-n_c}$

$\bar{D}$ number of parameters of complete probabilistic model

D number of parameters of predictor model

$\mathbf{D}(q) = (\mathbf{D}_1(q), \mathbf{D}_2(q))$ where

$\mathbf{D}_1(q) = d_{1,0} + d_{1,1} q^{-1} + d_{1,2} q^{-1} + \cdots + d_{1,n_{v1}} q^{-n_{v1}}$,

$\mathbf{D}_2(q) = d_{2,0} + d_{2,1} q^{-1} + d_{2,2} q^{-1} + \cdots + d_{2,n_{v2}} q^{-n_{v2}}$

$\mathbf{E}_k(q) = e_{k,0} + e_{k,1} q^{-1} + e_{k,2} q^{-2} + \cdots + e_{k,k-1} q^{-k+1}$

$\bar{f}$ joint PDF model of specific output sequence

f_0 true model between inputs and outputs

$\bar{f}_0$ true joint PDF model of specific output sequence

f_e PDF of prediciton error

$\mathbf{f}'(t)$ filtered response due to past input, past output, and past & future measurable disturbance sequences, i.e.,

$\mathbf{H}\Delta\tilde{\mathbf{V}}(t) + \mathbf{f}(t)$

$\mathbf{f}(t)$ free response, i.e.,

$\mathbf{G}_{\mathrm{p}k}(q)\Delta u(t-1) + \mathbf{H}_{\mathrm{p}k}(q)\Delta v(t) + (\mathbf{M}_k(q)\mathbf{F}_k(q) + \mathbf{N}_k(q))y(t)$

$\mathbf{F}_k(q) = f_{k,0} + f_{k,1}q^{-1} + f_{k,2}q^{-2} + \cdots + f_{k,n_a-1}q^{-n_a}$

$g(\cdot,\cdot;\theta)$ predictor model parameterised by θ

$\mathbf{G}_{\mathrm{f}k}(q) = g_{\mathrm{f}k,0} + g_{\mathrm{f}k,1}q^{-1} + g_{\mathrm{f}k,2}q^{-2} + \cdots + g_{\mathrm{f}k,k-1}q^{-k+1}$

$\mathbf{G}_{\mathrm{p}k}(q) = g_{\mathrm{p}k,0} + g_{\mathrm{p}k,1}q^{-1} + g_{\mathrm{p}k,2}q^{-2} + \cdots + g_{\mathrm{p}k,k+n_b-3}q^{-k-n_b+3}$

G_t discounted return

$G_{t:t+n}$ n-step return

G_t^λ λ-step return

$\tilde{h}_l$ hidden layer size, i.e., number of neurons of lth layer

$\mathbf{H}_{\mathrm{f}k}(q) = (\mathbf{H}_{\mathrm{f}k,1}(q), \mathbf{H}_{\mathrm{f}k,2}(q))$ where

$\mathbf{H}_{\mathrm{f}k,1}(q) = h^1_{\mathrm{f}k,0} + h^1_{\mathrm{f}k,1}q^{-1} + h^1_{\mathrm{f}k,2}q^{-2} + \cdots + h^1_{\mathrm{f}k,k-1}q^{-k+1}$,

$\mathbf{H}_{\mathrm{f}k,2}(q) = h^2_{\mathrm{f}k,0} + h^2_{\mathrm{f}k,1}q^{-1} + h^2_{\mathrm{f}k,2}q^{-2} + \cdots + h^2_{\mathrm{f}k,k-1}q^{-k+1}$

$\mathbf{H}_{\mathrm{p}k}(q) = (\mathbf{H}_{\mathrm{p}k,1}(q), \mathbf{H}_{\mathrm{p}k,2}(q))$ where

$\mathbf{H}_{\mathrm{p}k,1} = h^1_{\mathrm{p}k,0} + h^1_{\mathrm{p}k,1}q^{-1} + h^1_{\mathrm{p}k,2}q^{-2} + \cdots + h^1_{\mathrm{p}k,k+n_{d1}-2}q^{-k-n_{d1}+2}$,

$\mathbf{H}_{\mathrm{p}k,2} = h^2_{\mathrm{p}k,0} + h^2_{\mathrm{p}k,1}q^{-1} + h^2_{\mathrm{p}k,2}q^{-2} + \cdots + h^2_{\mathrm{p}k,k+n_{d2}-2}q^{-k-n_{d2}+2}$

$\mathcal{I}_{\cdot,\cdot}$ indecator function

K_d derivative gain of PID control

K_i integral gain of PID control

K_p proportional gain of PID control

L initial time index of manipulated input of MFAC

$\mathbf{M}_k(q) = m_{k,0} + m_{k,1}q^{-1} + d_{k,2}q^{-2} + \cdots + m_{k,k-1}q^{-k+1}$

N sample size

$\mathbf{N}_k(q) = n_{k,0} + n_{k,1}q^{-1} + n_{k,2}q^{-2} + \cdots + n_{k,n_c-1}q^{-n_c+1}$

N_r terminal time of output prediciton of GPC

q_π action-value function of an agent

$q_{\pi_i,i}$ action-value function of agent i

$Q(s,a)$ estimated value of action-value for state-action pair (s,a)

$\mathcal{R}$ set of rewards for an agent

r a reward

$\bar{r}(t)$ setpoint of carbon capture level

$\bar{\mathbf{Y}}_r(t)$ stacked reference signal vector, i.e.,

$(y_r(t+1), y_r(t+2), \cdots, y_r(t+N_r))^T$

$\mathcal{S}$ set of non-terminal states for an agent

$\mathcal{S}^+$ set of states including terminals for an agent

$\mathcal{S}_i$ set of states for agent i

$\bar{\mathbf{U}}(t)$ stacked input vector of future, i.e.,

$(u(t), u(t+1), \cdots, u(t+N_r-1))^T$

$\tilde{\mathbf{U}}(t)$ stacked input vector of past, i.e.,

$(u(t), u(t-1), \cdots, u(t+L-1))^T$

$\mathbf{u}^t$ input sequence till time t, $\{\mathbf{u}(0), \mathbf{u}(1), \cdots, \mathbf{u}(t)\}$

v_π state-value function of an agent

$\bar{\mathbf{V}}(t)$ stacked disturbance vector of future, i.e.,

$(v^T(t+1), v^T(t+2), \cdots, v^T(t+N_r))^T$

$\tilde{\mathbf{V}}(t)$ a variation of $\bar{\mathbf{V}}(t)$, i.e.,

$(v_1(t+1), v_1(t+2), \cdots, v_1(t+N_r), v_2(t+1), v_2(t+2), \cdots, v_2(t+N_r))^T$

$\mathbf{W}^l$ weight matrix of layer l for neural network

$\mathbf{x}(t)$ input feature vector

$\bar{\mathbf{Y}}(t)$ stacked output prediciton vector, i.e.,

$(\hat{y}(t+1|t), \hat{y}(t+2|t), \cdots, \hat{y}(t+N_r|t))$

y^t output sequence till time t, $\{y(0), y(1), \cdots, y(t)\}$

$z_t(s,a)$ eligibility trace for state-action pair (s,a)

$\mathbf{z}^l(t)$ output sequence till time t, $\{y(0), y(1), \cdots, y(t)\}$

$\alpha_t(s,a)$ learning rate of temporal difference learning for state-action pair (s,a) at time t

α_t learning rate of temporal difference learning for any state-action pair at time t

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

δ temporal difference error

$\delta(\cdot)$ degree of polynomial

$\varepsilon_t(s)$ probability of choosing action uniformly random for state s at time t

ε_t probability of choosing action uniformly random for any state at time t

ϵ prediction error

η tuning parameter of MFAC

γ discount rate of return

λ decay factor for λ return

$\tilde{\lambda}$ tuning parameter of MFAC

μ tuning parameter of MFAC

Φ pseudo-partial derivative vector, i.e., $(\phi_1, \phi_2, \cdots, \phi_L) \in \mathbb{R}^{1\times L}$

$\hat{\Phi}$ estimate of pseudo-partial derivative vector Φ, i.e., $(\hat{\phi_1}, \hat{\phi_2}, \cdots, \hat{\phi_L}) \in \mathbb{R}^{1\times L}$

π policy

$\boldsymbol{\rho}$ tuning parameter of MFAC

σ standard deviation of prediciton error

$\sigma_{\mathrm{a}}(\cdot)$ activation function of neural networks

θ parameter vector of predictor model

ϑ parameter vector of complete probabilistic model

$\|\cdot\|$ 2-norm of a vector or matrix

2. Dynamics of carbon capture plant

u lean MEA flow rate (kg/s)

$\mathbf{u}$ input vector for model identification, i.e., $(u, v_1, v_2)^T$

v disturbance vector, $(v_1, v_2)^T$

v_1 flue gas flow rate (kg/s)

v_2 mass fraction of CO_2 in the flue gas (unitless)

y carbon capture level (unitless)

$\hat{y}$ one-step-ahead prediciton of carbon capture level (unitless)

3. Parameters of power plant with carbon capture

B cost of CO_2 allowance bidding (€)

C annual expenditure (€)

$\mathcal{E}$ specific CO_2 emission ($\mathrm{kg_{CO_2}/GJ}$)

F fuel cost (€)

f fuel price (€/GJ)

F_{cc} fixed OM cost of carbon capture plant (€)

h CO_2 allowances in one's holding account (allowances)

$\bar{h}$ the upper holding limit of CO_2 allowances (allowances)

$\mathbb{I}$ set of all agents

$\mathcal{K}$ common knowledge set of a CO_2 allowance auction

$\mathcal{M}$ auction mechanism

$\dot{m}_{\mathrm{B}}$ base mass flow rate of CO_2 of a power plant without carbon capture (kg/s)

m emitted CO_2 (t)

$\mathcal{N}$ specific non-fuel OM cost (€/MWh)

N_{p} non-fuel OM cost (€)

N_{s} number of learning steps for an agent

$\mathcal{P}$ specific power usage within power plant (MW/MW)

p_{r} reserve price (€/GJ)

p_{a} penalty of non-compliance of CO_2 allowance surrendering (€/allowance)

p_{b} bid price (€/allowance)

$\mathcal{P}_{\mathrm{cc}}$ specific carbon capture plant load ($\mathrm{MWh/t_{CO_2}}$)

P_{g} net power output (MW)

P_{gB} nominal power capacity (MW)

P_{gg} gross power output (MW)

p_{s} settlement price (€/allowance)

q_{b} bid quantity (allowances)

q_{s} winning bid quantity (allowances)

$\mathcal{Q}_{\mathrm{reb}}$ specific reboiler duty ($\mathrm{MJ/kg_{CO_2}}$)

R_{E} revenue of a power plant with carbon capture (€)

M concerned time horizon of power plant with carbon capture (fortnight)

T_{a} period of the CO_2 allowance auction (days)

$\mathcal{V}_{\mathrm{cc}}$ specific variable OM cost of carbon capture plant (€/$\mathrm{t_{CO_2}}$)

v_{a} scheduled allowance volume for an auction

$\tilde{v}_{\mathrm{a}}$ allowance volume distributed to an auction due to failures of historical auctions

V_{cc} variable OM cost of carbon capture plant (€)

$\mathbf{y}$ allocations of all agents

η power efficiency without carbon capture (unitless)

λ_{E} average electricity price (€/MWh)

ψ power penalty coefficient due to carbon capture

The University of Manchester

Ziang Li
Doctor of Philosophy
Operation Issues on Carbon Capture Processes of Power Plants
June 5, 2019

Electricity and heat production contribute to about half of the CO_2 emission worldwide where fossil-fuel power plants are large emission sources. To reduce CO_2 emitted by these power plants, the mono-ethanol-amine-based post-combustion carbon capture technology can be utilised. Nevertheless, the integration of this technology with a fossil-fuel power plant causes non-trivial operation issues. In this thesis, some of these operation issues are solved from both the low-level control and the high-level decision-making perspectives. Proportional-integral control, generalised predictive control and model-free adaptive control protocols are implemented for the control issue while Sarsa and Sarsa(λ) learning algorithms are applied to the decision-making issue.

To demonstrate the performances of the control protocols, a neural-network-based carbon capture plant model using mono-ethanol-amine solvent is identified first. The model can pass the residual analysis and fit well with the data set generated by a validated first-principle post-combustion carbon capture model using mono-ethanol-amine solvent. On that basis, proportional-integral control, generalised predictive control and model-free adaptive control are implemented to solve the carbon capture level tracking issue. As demonstrated in the simulation results, generalised predictive control is difficult to implement. Proportional-integral control can be applied easily but with non-trivial tuning procedures. The model-free adaptive control is more robust than the proportional-integral control and has a simpler parameter tuning procedure than the other control protocols.

Apart from the control issue, the decision-making issue of the carbon capture level is addressed with either the Sarsa or Sarsa(λ) learning algorithm. In this thesis, the mono-ethanol-amine-based post-combustion carbon capture plants are also the participants in CO_2 allowance auctions. This necessitates a unified bidding and operation policy considering the time-varying operating conditions of the associated power plant, the electricity market conditions, and the CO_2 allowance auctions. The objective is to maximise the profits of any mono-ethanol-amine-based post-combustion carbon capture plant. Performances of the Sarsa and Sarsa(λ) learning algorithms are demonstrated in the simulation results which conclude that Sarsa should be preferable if data for learning can be cheaply generated and there exist no delayed penalties. Otherwise, Saras(λ) should be favoured but with more computation burdens.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library's regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University's Policy on Presentation of Theses.

Publications

1). Z. Li, Z. Ding, M. Wang, and E. Oko, "Model-free adaptive control for MEA-based post-combustion carbon capture processes," *Fuel*, vol. 224, pp. 637-643, 2018.

2). Z. Li, Z. Ding, and M. Wang, "Operation and bidding strategies of power plants with carbon capture," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3244-3249, 2017, 20th IFAC World Congress.

3). Z. Li, Z. Ding, and M. Wang, "Optimal bidding and operation of a power plant with solvent-based carbon capture under a CO_2 allowance market: A solution with a reinforcement learning-based sarsa temporal-difference algorithm," *Engineering*, vol. 3, no. 2, pp. 257-265, 2017.

Acknowledgements

This work would not have been possible without the help of many people.

First and foremost, I would like to thank Prof. Zhengtao Ding for his supervision. He encouraged and helped me a lot all through this four-year research period. He gave me valuable suggestions and guided me through some useful tools such as data-driven control and reinforcement learning.

Also, I would like to thank Prof. Meihong Wang and Dr. Eni Oko, The University of Sheffield. They gave me helpful recommendations and made great efforts to prepare some data for this research.

My thanks should also go to all the colleagues and staff of Control System Centre, The University of Manchester. They helped me to pass the research time joyfully.

Finally, I owe many thanks to my parents due to not only their financial support for my research but also emotional inspirations for my Manchester life from overseas.

Chapter 1

Introduction

1.1 Background

Carbon dioxide emission of power generation is a crucial factor contributing to global warming. According to the UK Future Energy Scenarios [7], before 2030, novel technologies tend to be demonstrated and invested to achieve commercial deployment of fossil-fuel power plants with carbon capture and storage (CCS). Pre-, post- and oxy-fuel combustions [4, 8, 9] are some promising state-of-the-art technologies which can be applied to carbon capture of carbon-intensive power plants. For the pre-combustion technology, syngas, a mixture of hydrogen, carbon monoxide and carbon dioxide, is generated from fossil fuels or biomass. Afterwards, the carbon monoxide is further converted into carbon dioxide and separated from the syngas where only pure hydrogen is retrieved for power generation. For the post-combustion technology, mono-ethanolamine (MEA) is used as the solvent for CO_2 removal from the flue gas emitted by a fossil-fuel power plant. Manzolini et al. [3] estimated the cost of CO_2 avoided for the amine-based post-combustion carbon capture (PCC) plants, which stated that coal-fired and natural gas power plants with MEA-based carbon capture were more cost-effective than power plants using renewable technologies. However, the regeneration of the MEA solvent with lean CO_2 loading is an energy-intensive process which constrains the electricity production of power plants. This issue necessitates the flexible control of carbon capture level in order to dynamically reduce energy consumption.

Last but not least, for the oxyfuel combustion technology, pure oxygen is used instead of air for a power plant, which yields some flue gas with high-concentration of CO_2. Nevertheless, the extraction of O_2 from the air will cause great energy penalty. Among the preceding three carbon capture technologies, this thesis focuses on the MEA-based post-combustion carbon capture. Some control and decision-making issues related to such a process are investigated, analysed, and solved via some state-of-the-art measures.

To begin with, it is worthwhile noting that the MEA-based PCC [10] technology is feasible for the large-scale CO_2 absorption since it not only leads to less CO_2 avoidance cost than the renewable but also can be achieved with relatively simple retrofits of conventional fossil-fuel power plants [11]. Nevertheless, one issue similar to the oxyfuel combustion technology is the significant energy penalty, which is about $3.2-3.6$ GJ/ton [12, 13]. Due to such a penalty, if the MEA-based PCC plant is integrated with a fossil-fuel power plant, some steam originally used for power generation is drawn off for carbon capture, which temporarily reduces the maximum output of the power plant. When there is some urgent power demand, it will be necessary to reduce the carbon capture level (i.e., the percentage of CO_2 which should be captured from the flue gas of a fossil-fuel power plant) or even stop the carbon capture plant swiftly to increase the power output. Therefore, a control protocol implemented on the carbon capture plant must have fast responses when the power demand soars up.

Furthermore, most fossil-fuel power plants contribute to base supplies in a power grid. Due to the consecutive large-scale penetrations of intermittent renewable power sources, a fossil-fuel power plant usually supplies flexible power generation and sometimes serves as a swing generator to compensate load variations. These may cause fluctuations of the emitted flue gas flow rate and the mass fraction of CO_2 in the flue gas which are external disturbances [14] of the MEA-based PCC process. These time-varying disturbances shift the operating points of the nonlinear plant back and forth and therefore deteriorate control performances. A control protocol for the process must be robust when confronting these uncertainties. When some tight CO_2 capture level should be met, the plant controller should be appropriately designed to ensure that the closed-loop system has robust responses.

Finally, a power plant is a for-profit entity. A carbon capture plant integrated with such a power plant, should operate with some specified profit-maximum objective as well. For instance, the carbon capture plant should determine the carbon capture level considering not only the operation of the carbon capture plant itself but also the market signals such as the electricity and CO_2 allowance prices [6, 15, 16, 17] in some optimal manner. When the electricity price is high or the CO_2 allowance price is relatively low, to meet the emission target of the corresponding power plant, the carbon capture plant should make proper decisions on whether to reduce the carbon emission with its carbon capture plant or procure more allowances through any applicable cap and trade program [18, 19]. In the thesis, the decision maker is only responsible for the operation and bidding of a carbon capture plant. Although this decision maker may coincide with the one of a power plant, in this thesis, we consider no electricity supply bidding under the wholesale electricity market. The MEA-based PCC plant only responds to some predefined power generation and electricity price profiles.

1.2 Motivations

The motivation of this research comes from two folds: The controller implemented to the MEA-based PCC plant must have fast and robust responses; the set-point of the carbon capture level should be optimally determined by the plant considering time-varying market conditions, e.g., the volatile electricity and CO_2 allowance prices.

For the flexible control issues of the carbon capture process, previous studies concentrated on the conventional proportional-integral (PI) control protocols [14, 20, 21, 22]. Nevertheless, for the carbon capture plant which is highly nonlinear, gain scheduling of those controllers should be applied for different operating regions of the plant, which is a non-trivial procedure. Recently, some model-based control protocols such as internal model control [20, 23] and model predictive control [24, 25, 26, 27, 28] are more favourable due to the requirements of fast and optimal responses, respectively. Nonetheless, these protocols make it mandatory to have an offline model through sensitivity or identification test for different operating regions, which is also time-consuming. In this thesis, we consider the novel model-free protocol, termed as

model-free adaptive control (MFAC) [29, 30, 31, 32]. This protocol is prevalent since an online model is already included in the control protocol and it adapts to changes of the environment recursively. As a result, the nonlinearity of the plant can be represented by updating the model parameters called the pseudo-partial derivative (PPD) vector/matrix, which implies MFAC is a robust control protocol. Additionally, in the following chapters, it is demonstrated that MFAC can achieve some fast responses similar to PI control and model predictive control (MPC) as well.

For the optimal operation of MEA-based PCC plant, most previous studies focus on the optimality of the plant itself without considering the market conditions. The carbon capture level for the tracking problem above, thus, is tightly set as some constant value, e.g., 90% [33]. There are three facts which make such a constant set-point unrealisable. Firstly, the solvent-based post-combustion carbon capture is an energy-intensive process. When the power demand is high, the associated fossil-fuel power plant should operate at its full capacity, under which circumstance, no extra energy can be used for carbon capture. Secondly, the previous literature usually assign a cost-minimum task rather a profit-maximum one for an MEA-based PCC plant while the commercial deployment of such a plant must ensure profitability under different market conditions. Some critical factors related to the economic operation of a carbon capture plant are the electricity and CO_2 allowance markets [16, 17, 34, 35], which highlights the significance of the market-oriented operation [36, 37, 38]. Thirdly, the market information taken into account in previous works are a predefined electricity price profile and a constant CO_2 emission tax. However, the main tool to dealing with the climate change in Europe should be the CO_2 emission trading system where most of the CO_2 allowances should be allocated through auctions. In this thesis, referring to the EU Emissions Trading System, we consider the CO_2 allowance auctions where the participants are the carbon capture plants integrated with fossil-fuel power plants. These plants can achieve their greenhouse gas emission target by bidding CO_2 emission allowances via auctions, directly reducing CO_2 with the integrated carbon capture plants or both of them. Whatever the policy is chosen by a plant, the objective is to maximise the plant profit. Reinforcement-learning algorithms, such as Sarsa and Sarsa(λ) [36, 37, 39] are applied to searching optimal policies.

1.3 Contributions

In the literature, one big challenge for the solvent-based carbon capture processes is to find a control protocol which can both perform well and be implemented easily. Model-based control, e.g., internal model control and model predictive control [20, 24, 40] are prevalent since they usually give better performances due to the inclusion of models. However, it also implies non-trivial model identification technologies are required for control implementation. On the other hand, one benefit of the PI control sometimes ignored by researchers is that it can be implemented easily. The reason is that PI control is model-free. No model identification is necessary for control implementation. Nonetheless, classic PI control usually performs worse than model-based control. In this thesis, we apply some other controller, i.e., MFAC which has the model-free characteristic but performs better than PI control, so as to give some insights to solve this challenging issue.

Another challenge is around the economic operation of the carbon capture plant [3, 41]. Most previous literature only focus on the economic operation issues in terms of the fixed carbon capture level, e.g., 90 %. However, the carbon capture level should be determined dynamically on the ground of the electricity and CO_2 allowance market conditions. In this thesis, we formulate the carbon capture problem considering emission trading system which is very crucial for the deployment of carbon capture technology. Provided that such kinds of technologies are likely to be commercialised, market price changes for CO_2 allowances will be the incentives which may greatly change the operation strategies of associated fossil-fuel power plants. Therefore, we present some possible model-free learning methods such as Sarsa and Sarsa(λ) which can be applied to find the operation strategies under the CO_2 allowance auction market.

Throughout this thesis, the model-free methodologies are applied to both the control and decision making of the MEA-based carbon capture plant integrated with a fossil-fuel power plant, which is the main contribution. Henceforth, the operation issues of the MEA-based PCC plant can be solved under the unified model-free framework. The detailed contributions of the thesis are listed as follows.

- A unified route map of the model identification, model order selection, and model validation for the MEA-based PCC plant is given under the neural-network-based model structures with data generated from a first-principle PCC plant model using MEA solvent [42, 43].

- A novel model-free control protocol termed as MFAC is implemented on the identified neural-network-based model.

- In comparison with some other conventional process control protocols, namely, PI control and generalised predictive control (GPC), the robustness of MFAC is demonstrated when confronting model uncertainties due to disturbances induced by the associated fossil-fuel power plant.

- A model of the CO_2 allowance auctions is constructed, through which, a profitability problem considering both the determination of carbon capture levels and the bidding of CO_2 allowances is formulated for MEA-based PCC plants.

- Model-free learning algorithms such as Sarsa and Sarsa(λ) are applied to optimally determining the operation and bidding policies of the MEA-based PCC plants integrated with fossil-fuel power plants which are also the participants in the preceding allowance auctions.

- The flexible operation of the MEA-based PCC plant is achieved with some model-free methodologies from both the low-level tracking and the high-level decision-making perspectives. The carbon capture levels are determined taking into account the bidding issue of CO_2 allowances in auctions.

1.4 Structure of the thesis

This thesis focuses on the control and decision-making issues of the MEA-based PCC plant integrated with a fossil-fuel power plant considering some electricity market conditions and CO_2 allowance auctions. The modelling and operation methodologies (for both control and decision making) of some literature are briefly introduced in Chapter 2. Afterwards, the related model-based and model-free control issues for the

carbon capture level tracking are investigated and extended in details in Chapter 3 and Chapter 4. In Chapter 5, the determination of the carbon capture levels is further discussed considering the fossil-fuel power plant operation, electricity market conditions and the CO_2 allowance auctions. This decision-making problem is solved based on some model-free algorithms termed as Sarsa and Sarsa(λ) learning algorithms. Conclusions are given in Chapter 6. Brief descriptions for each chapter are outlined as follows.

Chapter 1. The incentives and obstacles for the integration of MEA-based PCC plants to the conventional fossil-fuel power plants are discussed first. Afterwards, the motivations of our research are put forward while the associated contributions are listed. In the end, for a better understanding of the following chapters, the structure of the thesis is presented.

Chapter 2. The modelling methodologies of the carbon capture processes are reviewed in this chapter under both the first-principle and system identification frameworks. On that basis, the control and decision-making methodologies for the flexible operation of carbon capture process in the literature are reviewed. From the perspective of control, the existing protocols such as PI and MPC are introduced whereas some weaknesses of the previous control protocols, such as the complexity of tuning and the robustness, are discussed. From the perspective of decision making, the drawbacks of the previous operation policies are clarified. As an insight, the carbon capture level of a carbon capture plant is usually tightly set as a constant, which is not realistic when considering the time-varying market conditions of both electricity and CO_2 allowances.

Chapter 3. Model identification and corresponding model-based control of a carbon capture plant are demonstrated in this chapter. Neural networks are used as the model structures, under which, the model parameters are identified with the back-propagation algorithm. Subsequently, the model orders are selected in light of Akaike's information criterion (AIC) while the model validation is ensured through residual analysis. With the identified model functioned as the controlled plant, the model-based control, namely, GPC, is implemented on it. Performances of the GPC protocol are demonstrated within the controlled plant when there exist external disturbances due

to the time-varying operation of a fossil-fuel power plant. The concerned disturbances are the flue gas flow rate and the mass fraction of CO_2 in the flue gas.

Chapter 4. In this chapter, two model-free control protocols, PI control and MFAC are introduced. The derivation and implementation of the single-input-single-output (SISO) MFAC are explained in details. Thereafter, the extensions of MFAC to multi-input-multi-output (MIMO) cases are presented. The MFAC protocol is then implemented on the MEA-based PCC plant described by a neural-network-based model structure as above. The tracking problem of the carbon capture level is considered. For the comparison of the model-based and model-free control protocols, the simulation results give not only the performances of the PI control and the MFAC in this chapter but also the performances of the GPC discussed in the previous chapter.

Chapter 5. Some basics of dynamic programming and reinforcement learning are introduced in this chapter. Afterwards, instead of some constant carbon capture level in either Chapter 3 or Chapter 4, Sarsa or Sarsa(λ) learning algorithm is implemented to give a time-varying carbon capture level of the MEA-based PCC plant. The carbon capture levels are obtained based on time-varying operating conditions of the associated power plant, the electricity market, and the CO_2 allowance auctions. On those bases, power generations, electricity prices, allowance reserve prices, allowance volumes, and bidding behaviours of other MEA-based PCC plants are considered for the carbon capture level determination. The performances of Sarsa and Sarsa(λ) are compared and evaluated within a simulated model of the CO_2 allowance auctions where participants are a set of PCC plants.

Chapter 6. This is the conclusion part of the entire thesis where the critical results of this research are highlighted and possible future works are discussed.

Chapter 2

Literature review

2.1 Introduction

In this chapter, a literature review is given for previous studies on the dynamic control and steady-state decision-making issues of the MEA-based PCC plant. In addition, some basic knowledge on the modelling of the PCC process is introduced as well.

2.2 MEA-based post-combustion carbon capture

The first-principle modelling is an intriguing research field for the understanding of the MEA-based PCC process, which absorbs CO_2 in the flue gas emitted from a fossil-fuel power plant. The flow diagram of the CO_2 capture process is shown in Figure 2.1. Initially, the flue gas of a power plant is fed from the bottom of the absorber while the MEA solvent lean in CO_2 is injected from the top. After reactive absorption between the carbon dioxide and MEA in the column, the purified gas with less CO_2 is vented to the atmosphere while an MEA solvent rich in CO_2 is transferred into the cross heat exchanger. This rich MEA solvent is then preheated by the lean MEA solution also fed to the exchanger from the bottom of the stripper column. The absorber column is structured by corrugated packing sheets since they can expand the inner contact area between the flue gas and MEA solvents.

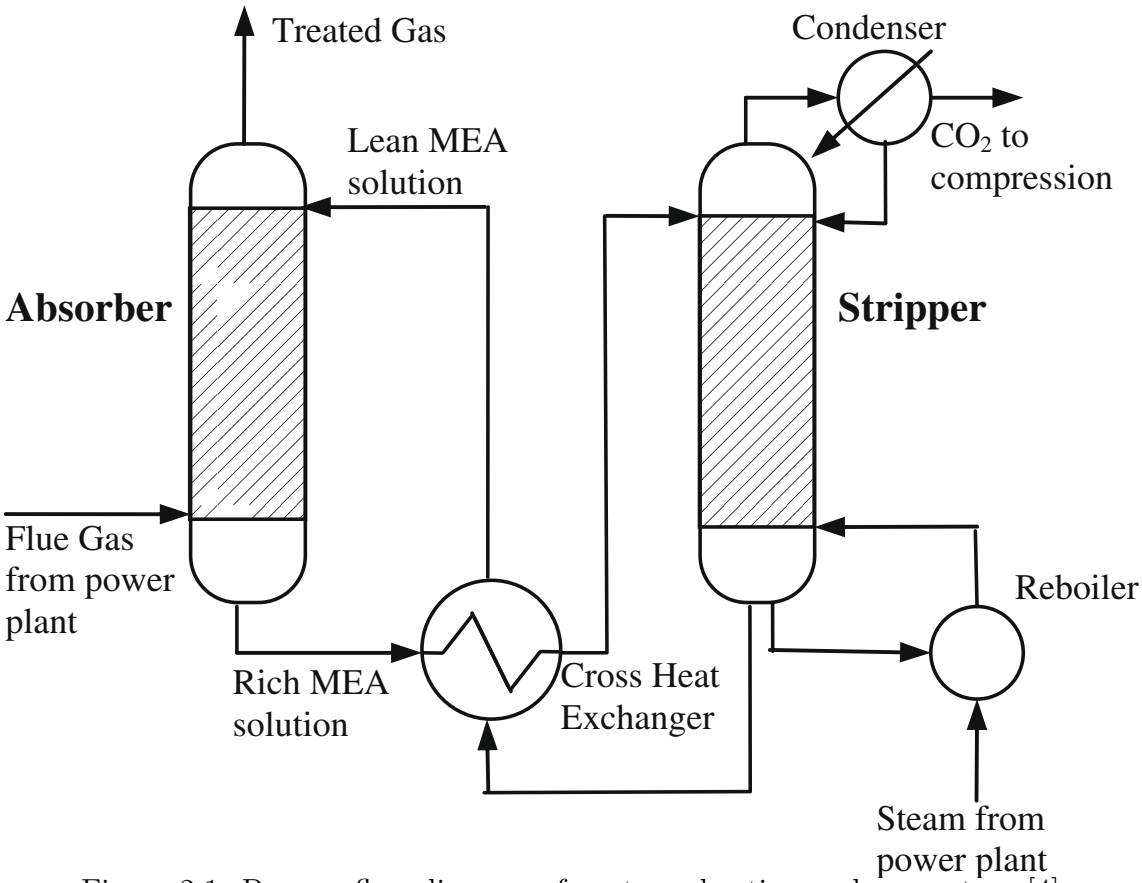


Figure 2.1: Process flow diagram of post-combustion carbon capture [4]

The stripper column has the analogous structure as the absorber. The pre-heated rich MEA is pumped to the upper-stage packing and then heated up when flowing down through the stripper unit for the regeneration of the lean MEA solvent. The heat is provided via a reboiler which is the primary energy consumption of the carbon capture process. Due to the rising solvent temperature, a mixed stream of the carbon dioxide and the water vapour will be extracted, which is further separated by a condensing unit downstream the stripper column. In the end, a high-purity CO_2 product stream is pumped out to some storage site.

To build a first-principle model of the MEA-based PCC process above, the mass transfer and chemical reactions in the absorber and stripper columns are the essential factors determining the modelling accuracy and complexity [20, 44, 45]. The columns are considered as cascaded small stages where the interaction occurs between the liquid and gas phases. From the perspective of the mass transfer, such interaction between the MEA solvent and the flue gas can be modelled with the two film theory

whereby the mass transfer is restricted to the two laminar film regions near the interface (Figure 2.2) [11]. In the previous literature, the equilibrium-based approach assumes an infinitely-fast mass transfer rate whereas the rate-based approach employs some feasible one, within those film regions.

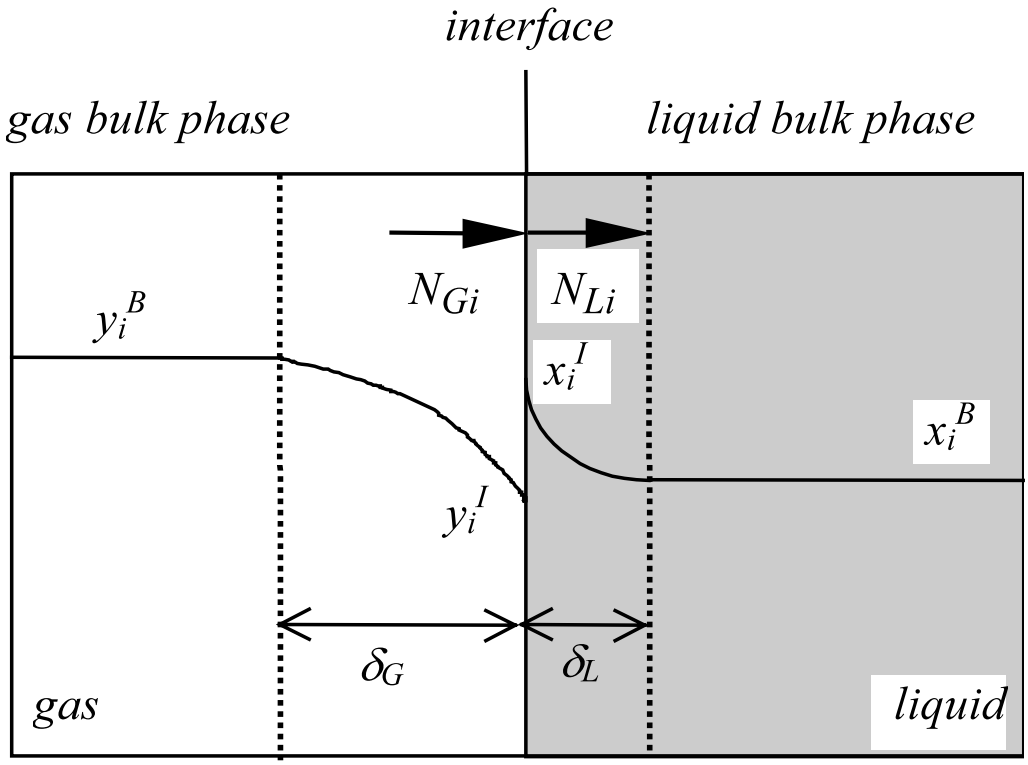


Figure 2.2: Two film theory[5]

A simple model will assume chemical reactions reaches equilibrium at each stage. To increase the model accuracy, some reaction kinetics can be included as well. Under the framework of the two-film-theory-based mass transfer, these chemical kinetics can be assumed to exist in just the bulk phase or both the bulk phases and film regions. These kinetic reactions [46] are represented as follows.

$$CO_2 + OH^- \longrightarrow HCO_3^- \text{ (kinetic)} \tag{2.1}$$

$$HCO_3^- \longrightarrow CO_2 + OH^- \text{ (kinetic)} \tag{2.2}$$

$$MEA + CO_2 + H_2O \longrightarrow MEACOO^- + H_3O^+ \text{ (kinetic)} \tag{2.3}$$

$$MEACOO^- + H_3O^+ \longrightarrow MEA + H_2O + CO_2 \text{ (kinetic)}. \tag{2.4}$$

Apart from those kinetic reactions [47], some simple equilibrium-based reactions should

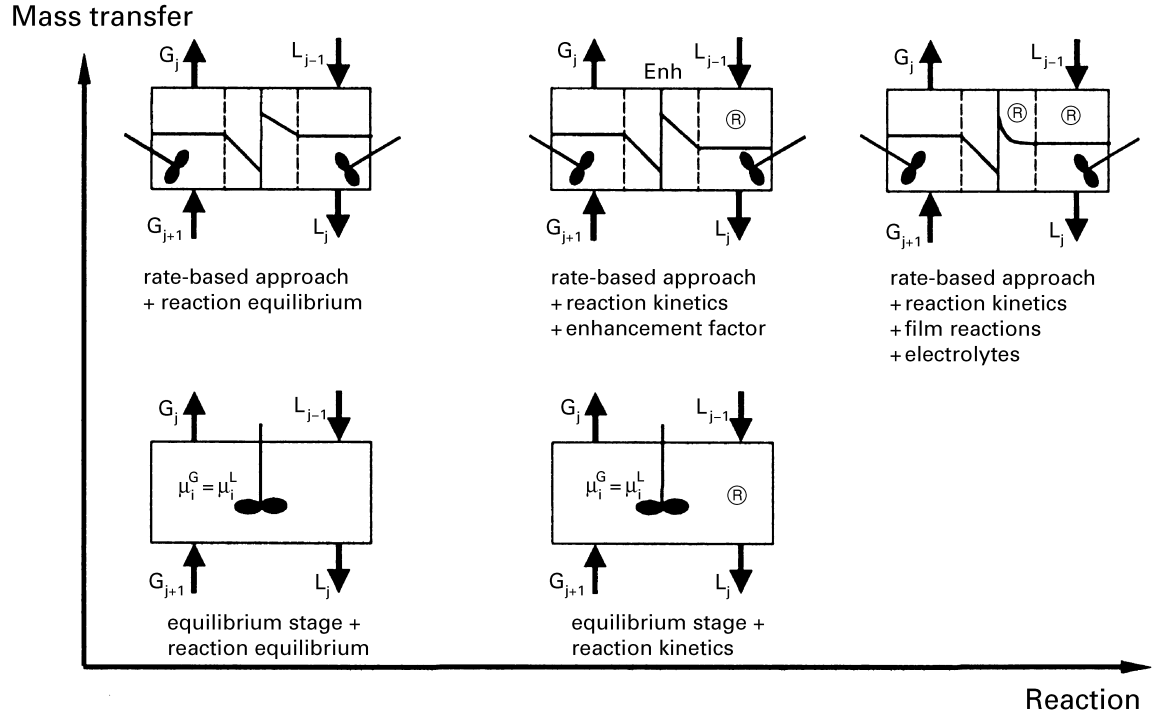


Figure 2.3: Model complexity [5]

be considered as well, i.e.,

$$H_2O + MEAH^+ \longleftrightarrow MEA + H_3O^+ \text{ (equilibrium)} \tag{2.5}$$

$$2H_2O \longleftrightarrow H_3O^+ + OH^- \text{ (equilibrium)} \tag{2.6}$$

$$HCO_3^- + H_2O \longleftrightarrow CO_3^{2-} + H_3O^+ \text{ (equilibrium).} \tag{2.7}$$

On the ground of the above discussion, the simplest absorber or stripper model is plotted in the bottom left of Figure 2.3 [5] where the stage is assumed to reach equilibrium in terms of both mass transportation and chemical reactions. Moving right forward gives a model considering chemical kinetics but still infinite-fast mass transportation. Since both of the bottom models neglect the mass transfer process, this set of modellings is usually called the equilibrium-based approach. On the contrary, three models on the top allow for the mass transfer rate, which is therefore termed as the rate-based approach. In addition, from the top left to the right, chemical kinetics in neither the film regions nor the bulk phases, the bulk phases, or both the film regions and the bulk phases, are considered with an increase of the model complexity.

Dynamic modelling methods of the PCC processes have been discussed in some previous literature [44, 48, 49]. According to Lawal et al. [44, 48], a rate-based approach

with equilibrium reaction is adopted for the modelling of an MEA-based PCC process whose performances are validated with the Case 32/47 data of a pilot plant [42]. In terms of the absorber temperature profile, it was illustrated that the rated-approach performed better than the equilibrium one. Additionally, the author highlighted the dominant influence of the liquid/gas ratio on the CO_2 removal percentage. According to Harun et al. [49], an overall description of the PCC process including a sump tank is presented. Tanks are ubiquitously embedded in an industrial-scale process and are essential when simulating a dynamic model. In this specific carbon capture process, one tank can be used for the storage of lean MEA solvent to bypass carbon capture temporarily while the other tank can be used for collection the rich MEA solvent when the electricity demand or price is high [10, 33].

Nevertheless, such a rate-based approach may be too complicated when it is served as the controlled plant for the model-based controllers such as MPC. A possible alternative is to construct one section of the PCC plant model at a time [10, 50, 51] or use some system-identification-based models [52, 53]. In this thesis, a validated model identified with the neural-network-based model structure is served as the simplified controlled plant of the preceding carbon capture process. This plant model can pass residual analyses and fit well with the data set generated by a validated first-principle dynamic model in gPROMS® [43].

2.3 Control issues of carbon capture processes

In some previous literature, PI control [14, 20, 21, 54, 55] is prevalent for the dynamic operation of the MEA-based PCC process since it can be implemented easily without any offline model. On the other hand, in terms of a customised performance index, MPC [24, 50, 56, 40, 57, 58] can achieve some better performances but requires an offline model for implementation. This underline model makes the design and tuning procedures of MPC non-trivial. For instance, if there exist mismatches between the model and the controlled plant, the performances of MPC will be degraded; if the model is too complex, the computation burden will make it impossible for the controller to respond the feedback signals in time. Most of the dynamic models for the

MEA-based PCC process are constructed in terms of the rigorous rate-based approach considering mass transfer rates [44]. Such a first-principle model is too complicated for the model-based control [29, 59]. To date, due to the model complexity, it is widely recognised that the first principle dynamic model with the rate-based approach and chemical reactions cannot be used for online control and optimisation. Some trade-off should be made so as to maintain the model accuracy but with less model complexity [10]. Such requirements can be achieved with sensitivity or identification tests [20, 60] but necessitate extra design and tuning efforts of controllers.

2.3.1 Model-free control

The previous model-free protocols applied to the MEA-based PCC process are PI controllers. Ziaii et al.[61, 62] revealed that the ratio control of the rich MEA flow rate by assuming a constant ratio between the rich solvent flow rate and the steam fed to the reboiler, under which circumstances, the disturbance due to the fluctuations of the steam flow rate to the reboiler could be compensated by the manipulations of the rich MEA flow rate. Lin et al. [21] constructed plant-wide decentralised control loops and pointed out the significance of the reboiler temperature control loop to reduce the energy consumptions for solvent regeneration. Nevertheless, for simplicity, only some equilibrium-based dynamic model was used to demonstrate their control performances. Wang et al. specified some decentralised control protocol of an MEA-based PCC plant for a coal-fired power plant [14, 63]. Their plant model was constructed with the rate-based approach and reaction reaching equilibrium. These papers introduced a detailed modelling of PCC process with decentralised control loops and then further integrated the carbon capture plant model with some coal-fired power plant model. It was highlighted that water make-up in the lean MEA solvent was critical to ensure a better performance of the carbon capture plant. In addition, integration of the carbon capture plant might change the transient dynamics of the associated fossil-fuel power plant due to the steam draw-off for lean MEA solvent regeneration. Nittaya et. al. [20] proposed three different PI control structures. The three structures are compared, which showed structure B was more appreciated which tracked the set-points (e.g., the carbon capture level) when facing the perturbations of the flue gas flow rate. They

also concluded that another control structure A designed by the relative gain array (RGA) analysis performed worse than structure B. One possible reason as pointed out could be that only the static sensitivity analysis rather than any dynamics was considered for the RGA-based loop selections.

As presented above, some literature have focused on the PI control protocols for the flexible operation of the MEA-based PCC processes. These protocols can be easily applied since no models are required for control implementation. However, to enhance the closed-loop performances, pairing strategies together with parameter tuning for each control loop should be decided carefully. Since a PCC plant usually behaves nonlinearly, gain scheduling should be given for different operating points of the controlled carbon capture plant.

2.3.2 Model-based control

For the model-based control, the problem focuses on minimising some performance index of the MEA-based carbon capture plant. Arce et al. [51] introduced a two-level MPC architecture whose performance indexes were operational costs and the tracking errors. For the high-level optimisation, costs of the CO_2 emission and the solvent-regeneration-based energy consumption were considered. For the low-level control, tracking errors of the CO_2 capture level and the reboiler liquid level were evaluated and then addressed by manipulating the solvent inlet flow and the power supply to the reboiler. Based on their control structure, a 10% reduction of energy cost for solvent regeneration was observed. Sahraei et al. [24] applied an MPC protocol for the carbon capture tracking issue. The robustness of the MPC is tested by adding step changes or oscillations of the flue gas flow rate. In comparison with the PI control, it was exhibited that MPC has a shorter settling time. Nonetheless, model linearisation is required to apply the MPC protocol since the nonlinear MPC had a much higher computation burden. Luu et al. [40] gave some systematic comparison of the performances for the standard proportional-integral-derivative (PID) feedback control, a cascade-PID control and MPC protocols. The MPC protocol was prevalent since

it could achieve some best performance subject to operational, economic and environmental constraints. Generally, these literature have focused on some performance indexes involving both tracking-error- and energy-consumption-based penalties. Nevertheless, some of them only considered these penalties with customised weightings while some others included some practical constant coefficients such as the electricity and CO_2 prices. As an extension, Arce et al. considered the some time-varying electricity price profile [51] for their MPC implementation.

As mentioned, some previous literature [16, 35, 40] have considered some CO_2 pricing mechanism but are based on the carbon tax or a predefined CO_2 price profile . Note that from 2013, power generators in Europe should bid allowances without free allocations through EU emission trading system (ETS). It implies some MPC protocol should consider the CO_2 allowance procurement from auctions.

2.4 Decision making issues of carbon capture processes

In the view of the economic performances of MEA-based PCC processes [16, 35, 64, 65, 66], it is essential to determine the optimal carbon capture levels under the electricity and CO_2 allowance market conditions. According to Abu-Zahra et al. [64, 67], the optimisation-based degrees of freedom for the standalone MEA-based PCC plant were the lean MEA loading, MEA concentration and the stripper operating pressure whose values are determined referring to no market conditions. The performance criteria were the consumption of cooling water and thermal energy (used for the lean MEA solvent regeneration). According to Mac Dowell et al. [35], the cost-effective lean loading for specified capture levels at the lowest reboiler heat duties (i.e., the main energy penalty for the system) was delivered, which was approximately 0.20 mol_{CO_2}/mol_{CO_2} [65, 68, 69]. Furthermore, not only lean loading but the capture level of the MEA-based PCC process should be manipulated to minimise total cost of carbon capture [17] taking into account the changing CO_2 prices in the market, which highlighted the market-oriented design and operation.

However, most of the relevant literature discussed the optimal operation of the MEA-based PCC plant under a cost-minimum target. For the commercial deployment, such a carbon capture process integrated with a fossil-fuel power plant, however, prefers to maximise its profits which are influenced by revenues of electricity output, cost of CO_2 allowances, as well as operation & maintenance (OM) expenditures. A fossil-fuel power plant should bid and then win adequate allowances from auctions to meet its demand for carbon dioxide emission. When the carbon capture plant is integrated with a fossil-fuel power plant, for a specified power load profile, decisions should be made on whether to capture CO_2 with the carbon capture plant or procure sufficient allowances from the CO_2 allowance auctions. Thus, the determination of the flexible carbon capture level should be considered together with the bidding problem of CO_2 allowances. Sarsa or Sarsa(λ) can be applied to search a unified optimal strategy which maximises profits during an evaluation period of the power plant.

2.4.1 Economic issues of carbon capture processes

For the optimal operation of the carbon capture plant, some operation parameters of the carbon capture plant should be optimised. Previous researches focus on the optimal operation of the carbon capture process alone. For instance, Abu-Zahra et al. [67, 64] gave the optimal operating conditions of the carbon capture plant in terms of the lean solvent loading, the amine solvent concentration, as well as the stripper operating pressure. Agbonghae et al. [65] gave some optimal lean CO_2 loading and liquid gas ratio for the MEA-based PCC plant integrated with either coal-fired or natural gas power plant considering the capital and operation costs. Subsequently, some economic issues such as the market conditions of electricity or CO_2 price are taken into consideration. According to Luo et al. [17], the levelised cost of electricity was applied as the objective to obtain required carbon price for the capture level of 90% under different fuel or transportation & storage prices. According to Mac Dowell et al. [35], the effect of the varying payback time for the optimal investment of a fossil-fuel plant with CCS was discussed when considering the lifetime-based cost.

However, the CO_2 price discussed in the literature above is more similar to the

carbon tax [70]. Influences of the flexible market clearing prices in terms of CO_2 allowance auctions are seldom investigated. On the other hand, although there exist researches on CO_2 allowance auction mechanisms [18, 71] and bidding strategies [72], they seldom mention the flexible operation of power plants with carbon capture systems. Besides, as in Subsection 2.3.2, power generators in principle should bid CO_2 allowances from auctions in Europe. The relationship between PCC operation and allowance bidding is important for the commercial deployment of the carbon capture plants within fossil-fuel power plants.

2.4.2 Emission trading system

EU ETS [73] is the key tool trying to set up a CO_2 allowance market to cope with the climate change problem in Europe. It sets a unified emission cap in each year for all the member states. In 2013, the unified emission cap is 2,084,301,856 allowances where one allowance is the permit of one ton CO_2 emission. Such a cap will account for over 45% of the EU greenhouse gas emission. Thereafter, the cap will decrease by 38,264,246 allowances/year (2013-2020) to meet the EU's emission target by 2020, i.e., 20% emissions reduction relative to 2005. For allowances scheduled in the cap, around 57% will be auctioned in the CO_2 allowance auctions while the rest will be freely allocated. Some typical CO_2 allowance settlement price is 15 €/allowance in 2018. From 2013, power generators should bid allowances through auctions. Adequate allowances should be bought from EU ETS to cover the fossil-fuel power plant greenhouse gas emission. Otherwise, non-compliance will cause an additional penalty of 100 €/allowance for each ton of excess CO_2 emission.

2.5 Summary

In this chapter, the MEA-based PCC process modelling using either the first-principle or system-identification-based methodologies are introduced. On that basis, some preferred control protocols in literature e.g., PI and MPC are discussed. Decision-making issue is reviewed thoroughly from both aspects. Based on previous literature,

some insights are given as follows.

- The first-principle-based model using the rate-based approach and reaction kinetics are too complex for the dynamic models with online control and optimisation.
- Model simplification is required for control-oriented modelling. Possible simplification methods include the construction of the equilibrium-based model rather than the rate-based ones, using system-identification-based dynamic models, or just focusing on the key part of the plant related to any specific control issue.
- Previous determination of the carbon capture level for the MEA-based PCC plant is trivial. It is tightly set as some constant, such as 90%.
- Due to the requirements of economically efficient operation, the different carbon capture levels should be evaluated in terms of different electricity and CO_2 prices.
- Since the power generators should bid CO_2 allowances without free allocations, the carbon capture level determination should be considered together with the CO_2 allowance auctions, which implies a unified bidding and operation problem should be formulated.

Chapter 3

Model-based control of MEA-based post-combustion carbon capture processes

3.1 Introduction

In this chapter, we first introduce some basics about how to build a proper mathematical model in terms of the dynamics of the carbon capture process using system identification methodology. Afterwards, an advanced model-based control protocol is implemented for the selected model. Assessments are made in terms of the performances of the model identification, model order selection and model-based control algorithm. The first-principle dynamic model of the PCC process has been developed in gPROMS® with the rate-based approach using the design and operation specifications due to the contributions of [43]. All the reactions in PCC are assumed to attain equilibrium. Validation of this model was made using data of pilot plants [14, 42]. The flow diagram (Fig. 3.1) shows the flue gas is initially fed into the bottom of the absorber while the lean MEA solution is injected from the top. After chemical reactions between CO_2 and the lean MEA countercurrently in the column, the purified gas with less CO_2 is vented to the atmosphere while a carbon-rich MEA solution is pumped into

the downstream lean/rich cross heat exchanger and exchanges energy with the lean solution from the stripper. The stripper has the analogous structure as the absorbers. The pre-heated rich MEA from the exchanger outlet is pumped to the upper-stage and heated up when flowing down through the column. The heat is provided via a reboiler which separates CO_2 from the rich MEA and reproduces the lean MEA to process the consecutively discharged flue gas. Although a rigorous model can be built considering chemical reactions, it is too complex for control design [59]. A feasible mathematical model must be identified, compared and validated [51]. Neural networks are such a powerful tool which can deal with the complexity issues. Nevertheless, in this thesis, since we mainly focus on the control (Chapters 3 and 4) and subsequent decision making (Chapters 5) issues, modelling should be general rather than accurate. Thus, we consider neural networks with one hidden layer for simplicity and try to validate their generality through the residual analysis in Subsection 3.2.5.

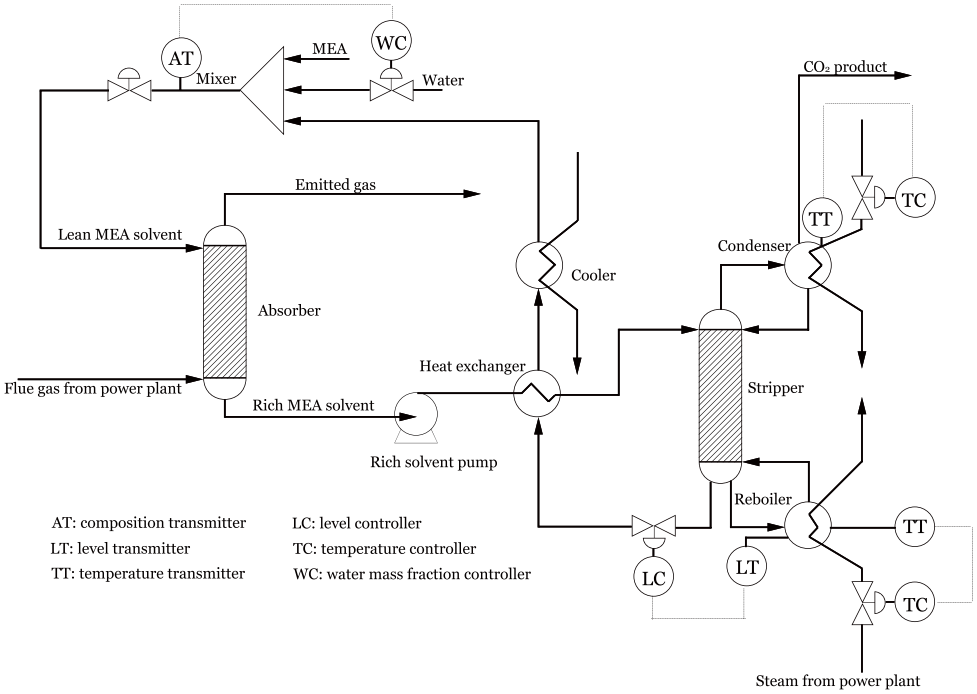


Figure 3.1: The process flow diagram of a PCC plant [6].

3.2 Model identification, order selection and validation

3.2.1 Preliminaries

The models considered in this chapter are complete probabilistic models [60] with three inputs $\mathbf{u} \in \mathbb{R}^3$ and one output $y \in \mathbb{R}$. A complete probabilistic model means a candidate model for the formulation of the carbon capture process should include both a predictor model and a probability density function (PDF) of the associated prediction errors. For simplicity, the predictor model here is supposed to achieve one-step-ahead prediction such that the output $y(t)$ can be predicted in terms of the sequences of historical inputs $\mathbf{u}^{t-1}$ and outputs y^{t-1},

$$\hat{y}(t|y^{t-1}, \mathbf{u}^{t-1}) \triangleq \hat{y}(t|t-1) = g(y^{t-1}, \mathbf{u}^{t-1}) \tag{3.1}$$

where

$$\mathbf{u}^t \triangleq \{\mathbf{u}(0), \mathbf{u}(1), \cdots, \mathbf{u}(t)\} \tag{3.2}$$

$$y^t \triangleq \{y(0), y(1), \cdots, y(t)\}. \tag{3.3}$$

Apart from the predictor model, the prediction error, is defined as

$$\begin{aligned} \epsilon(t) \triangleq & y(t) - \hat{y}(t|t-1) \\ = & y(t) - g(y^{t-1}, \mathbf{u}^{t-1}) \end{aligned} \tag{3.4}$$

which must be fully specified through a PDF as mentioned before. Furthermore, to search over a complete probabilistic model set (or simply, a model set) smoothly, $g(\cdot, \cdot)$ is parameterised by θ and expressed as $g(\cdot, \cdot; \theta)$. Such a parameterised set of models are termed as a model structure [60] for the one-step-ahead prediciton $\hat{y}(t|t-1)$. The parameterised predictor model and prediction error, thus, can be rewritten as

$$\hat{y}(t|t-1) = g(y^{t-1}, \mathbf{u}^{t-1}; \theta) \tag{3.5}$$

$$\epsilon(t) = y(t) - g(y^{t-1}, \mathbf{u}^{t-1}; \theta). \tag{3.6}$$

In this section, neural networks are assumed to be the aforementioned predictor models, i.e.,

$$g(y^{t-1}, \mathbf{u}^{t-1}; \theta) = \mathbf{a}^{\bar{l}}(t) = \mathbf{W}^{\bar{l}}\mathbf{a}^{\bar{l}-1}(t) + \mathbf{b}^{\bar{l}-1} \tag{3.7}$$

subject to

$$\mathbf{a}^{l}(t) \triangleq \sigma_{\mathrm{a}}(\mathbf{z}^{l}(t)) \tag{3.8}$$

$$\mathbf{z}^{l}(t) \triangleq \mathbf{W}^{l}\mathbf{a}^{l-1}(t) + \mathbf{b}^{l} \tag{3.9}$$

$$\mathbf{a}^{1}(t) = \mathbf{x}(t) \tag{3.10}$$

where $\mathbf{z}^{l}(t)$, $\mathbf{a}^{l}(t)$, $\mathbf{b}^{l} \in \mathbb{R}^{\tilde{h}_l}$, $\mathbf{x}(t)$ is the input feature vector of a neural network, $\tilde{h}_l$ is the layer size, i.e., the number of neurons of layer l, and $\sigma_{\mathrm{a}}(\cdot)$ is an element-wise activation function for each entry of $\mathbf{z}^{l}$. Without losing generality, for $\mathbf{z}^{l} = z \in \mathbb{R}$, the scalar activation function is logistic, i.e.,

$$\sigma_{\mathrm{a}}(z) = \frac{1}{1 + \exp(-z)}. \tag{3.11}$$

The superscript $l \in \{2, 3, \cdots, \bar{l} - 1\}$ indicates the layer index of a specified weight matrix $\mathbf{W}^{l}$ or bias vector $\mathbf{b}^{l}$. Correspondingly, an entry w_{ij}^{l} of $\mathbf{W}^{l}$ is the weight on the branch between the jth neuron of layer $l-1$ and the ith neuron of layer l while an entry b_{i}^{l} of $\mathbf{b}^{l}$ is the scalar bias of the ith neuron of layer l. It should be noted that when a layer has only one neuron, the weight matrix and bias vector may be reduced to a vector and scalar, respectively. For instance, since $y(t)$ is a scalar, $\mathbf{W}^{\bar{l}} = (w_{11}^{\bar{l}}, w_{12}^{\bar{l}}, \cdots, w_{1\tilde{h}_{\bar{l}-1}}^{\bar{l}}) \in \mathbb{R}^{1\times\tilde{h}_{\bar{l}-1}}$ and $\mathbf{b}^{\bar{l}} = b_{1}^{\bar{l}} \in \mathbb{R}$ of the output layer (i.e., $l = \bar{l}$) are a row vector and scalar, respectively. The objective function can be then defined as

$$\tilde{C}(\mathbf{W}^{2}, \mathbf{W}^{3}, \cdots, \mathbf{W}^{\bar{l}}, \mathbf{b}^{2}, \mathbf{b}^{3}, \cdots, \mathbf{b}^{\bar{l}}) = \frac{1}{2}\sum_{t=1}^{N} |\epsilon(t)|^{2}. \tag{3.12}$$

3.2.2 Problem formulation

For the PCC process which is complex and nonlinear, neural networks [52, 53] can be selected as the predictor model for identification based on the offline data generated by the first-principle carbon capture model. Note that the tracking problem of the carbon

capture level is primarily considered in this chapter. For brevity, lean CO_2 loading and the reboiler temperature are assumed to be fixed around 0.28 mol/mol and 387 K, respectively, for all cases in the later simulations of this chapter. On that basis, a model related to dynamics of the carbon capture level is built with three inputs

$$\mathbf{u}(t) \triangleq (u(t), v_1(t), v_2(t))^T \tag{3.13}$$

where $u(t)$, $v_1(t)$, and $v_2(t)$ are the lean MEA flow rate (kg/s), the flue gas flow rate (kg/s), and the mass fraction of CO_2 in the flue gas, respectively, and one output, namely, the carbon capture level (%), denoted by $y(t)$.

The concerned candidate predictor models in this chapter are neural networks with one hidden layer with varying sizes. Referring to Figure 3.2 and Eq. (3.7), these predictor models can be represented by

$$\hat{y}(t|t-1) = g(y^{t-1}, \mathbf{u}^{t-1}; \theta) = \mathbf{W}^3 \sigma_{\mathrm{a}}(\mathbf{W}^2 \mathbf{x}(t) + \mathbf{b}^2) + \mathbf{b}^3 \tag{3.14}$$

The input feature vector $\mathbf{x}(t) \in \mathbb{R}^{h_1}$ at time t is then defined as

$$\begin{aligned} \mathbf{x}(t) \triangleq & (y(t-1),\, y(t-2),\, \cdots,\, y(t-n_a), \\ & v_1(t-1),\, v_1(t-2),\, \cdots,\, v_1(t-n_{d1}), \\ & v_2(t-1),\, v_2(t-2),\, \cdots,\, v_2(t-n_{d2}), \\ & u(t-1),\; u(t-2),\; \cdots,\; u(t-n_b))^T \end{aligned} \tag{3.15}$$

with $\tilde{h}_1 = n_a + n_b + n_{d1} + n_{d2}$. n_a, n_b, n_{d1}, and n_{d2} are model orders which must be determined in terms of model performances. For a specific candidate model based on neural networks, the parameter vector of a predictor model (Eq. (3.14)), θ, is defined based on the weights $\{\mathbf{W}^2, \mathbf{W}^3\}$ and biases $\{\mathbf{b}^2, \mathbf{b}^3\}$, i.e.,

$$\theta \triangleq \mathbf{col}(\mathbf{W}^2, \mathbf{W}^3, \mathbf{b}^2, \mathbf{b}^3) \in \mathbb{R}^D, \tag{3.16}$$

where $\mathbf{col}(\cdot)$ means the entries of parameters are stacked and columnised as a vector. These model parameters should be identified using the input and output data from the first-principle model. The total number of entries of the predictor model parameters should equal to $D = [(\tilde{h}_1 + 2) \cdot \tilde{h}_2] + 1$. To avoid over-fitting [74], for two candidate models with similar model validation performances, the model with less complexity, i.e., smaller D, is preferred.

Apart from the prediciton model, the prediction errors $\{\epsilon(t)\}$ are supposed to be white Gaussian noises, i.e., a sequence of independent and identically distributed random variables with $\epsilon(t) \sim \text{Normal}(0, \sigma^2)$ expressed as

$$f_e(\epsilon(t)) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp[-\frac{|\epsilon(t)|^2}{2\sigma^2}]. \tag{3.17}$$

The parameterised set of models described by Eq. (3.14) and Eq. (3.17) is termed as a model structure. Note that under a specific model structure, the dimension of the parameter vector D is fixed. Thus, the best model can be selected with maximum likelihood estimate (MLE) [75], or the back-propagation algorithm for feed-forward neural networks in this case. Note that for a real system, we may be unable to represent the true model using an existing model structure or there just exists no true model. Nevertheless, all models are wrong while some are useful if they can pass validation tests. In this thesis, we assume that the prediction error should suffice Gaussian distribution. Then, the validation is made in terms of two test statistics, i.e., ζ_1 and ζ_2 introduced in Subsection 3.2.5. To ensure the assumption is valid, the statistics must suffice specified probability distributions. Theses statistics can be calculated based on the prediction errors, i.e., the residuals (Eq. (3.4)) of the predictor model (Eq. (3.14)).

3.2.3 Model identification

For the above neural-network-based predictor model (Eq. (3.14)) and prediction error (Eq. (3.6)), a cost function can be constructed as

$$\tilde{C}(\theta) = \tilde{C}(\mathbf{W}^2, \mathbf{W}^3, \mathbf{b}^2, \mathbf{b}^3) = \frac{1}{2}\sum_{t=1}^{N} |\epsilon(t)|^2. \tag{3.18}$$

which should be minimised with respect to (w.r.t.) weights and biases for model identification. A gradient descent algorithm can be applied with

$$\mathbf{W}^l \leftarrow \mathbf{W}^l - \tilde{\eta}\frac{\partial \tilde{C}}{\partial \mathbf{W}^l} \tag{3.19}$$

$$\mathbf{b}^l \leftarrow \mathbf{b}^l - \tilde{\eta}\frac{\partial \tilde{C}}{\partial \mathbf{b}^l} \tag{3.20}$$

where $l \in \{2, 3\}$. However, the computation burden of such an algorithm is significant when the sample size N is large. Usually, a mini-batch sample size N' (much less than

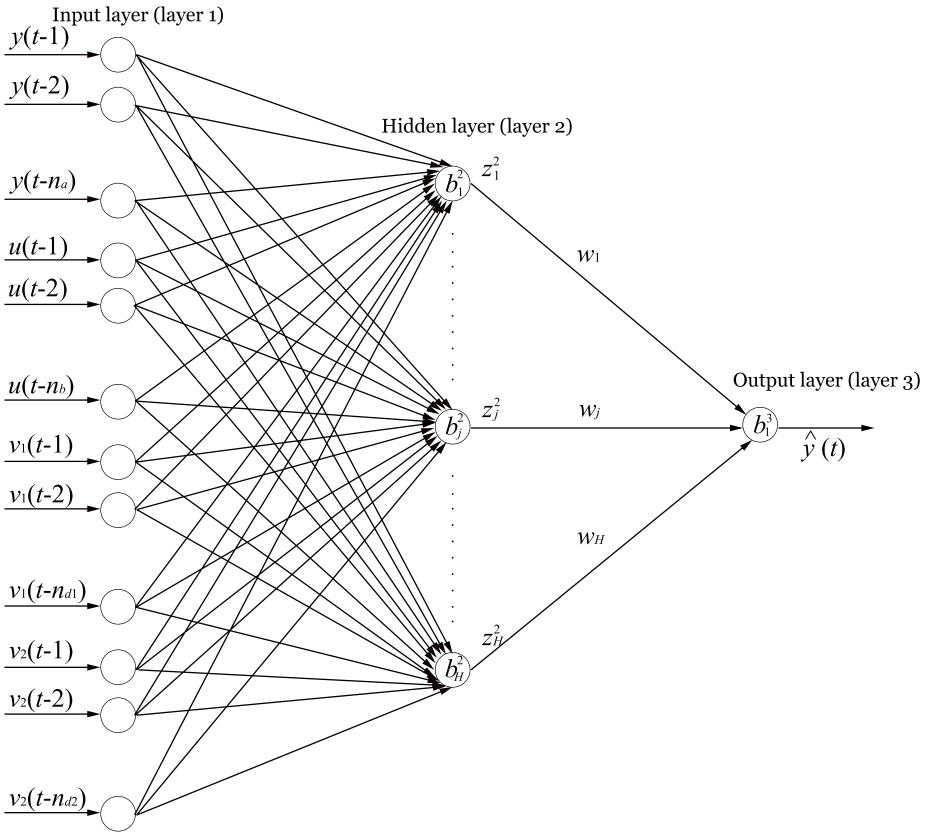


Figure 3.2: A multi-input-single-output neural network with one hidden layer.

N) should be specified to reduce time for the calculation of $\partial\tilde{C}/\partial\mathbf{W}^l$ and $\partial\tilde{C}/\partial\mathbf{b}^l$ [76]. When $N' = 1$, the cost function can be reduced to

$$\tilde{C}(\mathbf{W}^2, \mathbf{W}^3, \mathbf{b}^2, \mathbf{b}^3) = \frac{1}{2}|\epsilon(t)|^2 = \frac{1}{2}|y(t) - \mathbf{a}^3(t)|^2, \ \forall\ t \in \{1, \cdots, N\} \tag{3.21}$$

which can be solved by the stochastic gradient descent algorithm identical to Eq. (3.19) and Eq. (3.20). The stochastic gradient descent $\partial\tilde{C}/\partial\mathbf{W}^l$ and $\partial\tilde{C}/\partial\mathbf{b}^l$ can be updated through the classic algorithm as follows [77]:

$$\begin{cases} \tilde{\delta}^{\bar{l}} = (y - \mathbf{a}^{\bar{l}}) \\ \frac{\partial\tilde{C}}{\partial\mathbf{b}^{l+1}} = \tilde{\delta}^{l+1} \\ \frac{\partial\tilde{C}}{\partial\mathbf{W}^{l+1}} = \tilde{\delta}^{l+1} \cdot (\mathbf{a}^l)^T \\ \tilde{\delta}^l = ((\mathbf{W}^{l+1})^T \cdot \tilde{\delta}^{l+1}) \odot \frac{\partial\sigma_{\mathrm{a}}(\mathbf{z}^l)}{\partial\mathbf{z}^l} \end{cases} \tag{3.22}$$

with $l \in \{1, 2, \cdots, \bar{l} - 1\}$. Note that since l should be assigned with $\bar{l} - 1, \bar{l} - 2, \cdots, 1$ orderly and the gradients ($\partial\tilde{C}/\partial\mathbf{W}^{l+1}$ and $\partial\tilde{C}/\partial\mathbf{b}^l$) are computed in a back-propagated manner, the recursive update procedure (Eq. (3.22)) is termed as the back-propagation

algorithm. Subsequently, Eq. (3.19) and Eq. (3.20) can be used for the updates of $\mathbf{W}^l$ and $\mathbf{b}^l$ for $\forall l$. The back-propagation algorithm listed above is used for regression problems [78]. To solve classification problems, a user may have

$$\tilde{\delta}^{\bar{l}} = (y - \mathbf{a}^{\bar{l}}) \odot \frac{\partial \sigma_{\mathrm{a}}(\mathbf{z}^{\bar{l}})}{\partial \mathbf{z}^{\bar{l}}}$$

instead of the first equation in Eq.(3.22). It is worthwhile noting that this back-propagation algorithm is actually derived when the model structure is differentiable [60]. If there are several applicable model structures, for instance, when the dimension of the parameter vector varies, such an algorithm can determine the best model of each model structure but may not give direct assessment w.r.t. the decrements or increments of the parameter vector dimension D. Therefore, AIC is introduced as follows to solve the model order selection issues considering the curse of dimensions.

3.2.4 Model order selection

From Eq. (3.4) and Eq. (3.17), it can be implied that the probability of observing $y(t)$ given $(\mathbf{u}^{t-1}, y^{t-1})$ should be equivalent to

$$p(y(t)|y^{t-1}, \mathbf{u}^{t-1}) = f_e(y(t) - g(\mathbf{u}^{t-1}, y^{t-1}; \theta)). \tag{3.23}$$

Thus,

$$\begin{aligned} p(y(t), y(t-1)|y^{t-2}, \mathbf{u}^{t-2}) =& p(y(t)|y(t-1), y^{t-2}, \mathbf{u}^{t-2}) p(y(t-1)|y^{t-2}, \mathbf{u}^{t-2}) \\ =& f_e(y(t) - g(y^{t-1}, \mathbf{u}^{t-1}; \theta)) f_e(y(t-1) - g(y^{t-2}, \mathbf{u}^{t-2}; \theta)). \end{aligned} \tag{3.24}$$

By induction, one should have

$$\begin{aligned} p(y^t|\mathbf{u}^t) =& \prod_{k=1}^{t} f_e(y(k) - g(y^{k-1}, \mathbf{u}^{k-1}; \theta)) \\ =& \prod_{k=1}^{t} \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp[-\frac{1}{2\sigma^2}(y(k) - \hat{y}(k|k-1))^2] \\ \triangleq& \bar{f}(y^t|\mathbf{u}^t, \theta, \sigma) \triangleq \bar{f}(y^t|\mathbf{u}^t, \vartheta) \end{aligned} \tag{3.25}$$

where $\vartheta \triangleq (\theta^T, \sigma)^T \in \mathbb{R}^{\bar{D}}$ with $\bar{D} = D+1$ is the dimension of the parameter vector of a complete probabilistic model, for which, in addition to those parameters of a predictor

model (Eq. (3.14)), the parameters of the prediction error model (Eq. (3.17)) should be included as well. For the specific PDF of the prediction error, f_e (Eq. (3.17)) above, only σ is the extra parameter added to form the augmented parameter vector ϑ. Moreover, $\bar{f}(y^t|\mathbf{u}^t,\vartheta)$ is the joint PDF model which describes the probability of observing the output sequence y^t, given the deterministic input sequence $\mathbf{u}^t$.

A typical model selection tool, AIC [74], is on the ground of the Kullback-Leibler (KL) distance with the joint PDF model (Eq. (3.25)), i.e.,

$$I(\bar{f}_0,\bar{f}) = \int \bar{f}_0(y^N|\mathbf{u}^N)\log(\frac{\bar{f}_0(y^N|\mathbf{u}^N)}{\bar{f}(y^N|\mathbf{u}^N,\vartheta)})dy^N \tag{3.26}$$

where N is the sample size and $\bar{f}_0$ is the true joint PDF of the observed output sequence. Ideally, the best parameterised model $\bar{f}(y^N|\mathbf{u}^N,\vartheta)$ should equal to $\bar{f}_0(y^N|\mathbf{u}^N)$. As a result, $I(\bar{f}_0,\bar{f})=0$ which is the minima of KL distance. However, in practice, it is uncertain whether we include the true model in the candidate model sets. Therefore, instead of seeking $I(\bar{f}_0,\bar{f})=0$, one should find the model with the shortest KL distance, which means

$$\vartheta_0 = \arg\min_{\vartheta}\int \bar{f}_0(y^N|\mathbf{u}^N)\log(\frac{\bar{f}_0(y^N|\mathbf{u}^N)}{\bar{f}(y^N|\mathbf{u}^N,\vartheta)})dy^N \tag{3.27}$$

where ϑ_0 is the parameter vector which is the best under the specific model structure (Eq. (3.14) and (3.17)). MLE can be applied with large samples to find this best parameter vector ϑ. Nonetheless, such a best model can be found only stochastically under a large sample size, i.e., the estimate of ϑ_0, denoted by $\hat{\vartheta}$, is a random variable dependent of the observed samples, z^N. Hence, a user should find the estimated model parameters whose expectation is minimized, which is

$$\begin{aligned}
\hat{\vartheta} &= \arg\min_{\hat{\vartheta}}\int \bar{f}_0(z^N|\mathbf{u}^N)\int \bar{f}_0(y^N|\mathbf{u}^N)\log(\frac{\bar{f}_0(y^N|\mathbf{u}^N)}{\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))})dy^N dz^N \\
&= \arg\min_{\hat{\vartheta}}\{\mathbb{E}_{z^N}[\int \bar{f}_0(y^N|\mathbf{u}^N)\log\bar{f}_0(y^N|\mathbf{u}^N)dy^N] \\
&\quad - \mathbb{E}_{z^N}[\int \bar{f}_0(y^N|\mathbf{u}^N)\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))]dy^N]\} \\
&= \arg\max_{\hat{\vartheta}}\{\mathbb{E}_{z^N}[\int \bar{f}_0(y^N|\mathbf{u}^N)\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))]dy^N]\} \\
&= \arg\max_{\hat{\vartheta}}\{\mathbb{E}_{z^N}[\mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))]]]\}
\end{aligned} \tag{3.28}$$

where $\mathbb{E}_{z^N}[\int \bar{f}_0(y^N|\mathbf{u}^N)\log\bar{f}_0(y^N|\mathbf{u}^N)dy^N]$ is neglected since it is a constant. $\mathbb{E}_{x^N}(\cdot)$ stands for $\int \bar{f}_0(x^N|\mathbf{u}^N)(\cdot)dx^N$ with $x^N\in\{y^N,z^N\}$. It can be derived in terms of [74]

that

$$\mathbb{E}_{z^N}[\mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))]]] \approx \mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(\vec{y}))]] - \mathbf{tr}[J(\vartheta_0)I^{-1}(\vartheta_0)] \tag{3.29}$$

where

$$J(\vartheta) = \mathbb{E}_{x^N}[[\frac{\partial \log[\bar{f}(x^N|\mathbf{u}^N,\vartheta)]}{\partial \vartheta}][\frac{\partial \log[\bar{f}(x^N|\mathbf{u}^N,\vartheta)]}{\partial \hat{\vartheta}}]^T]$$

and

$$I(\vartheta) = -\mathbb{E}_{x^N}[\frac{\partial^2 \log[\bar{f}(x^N|\mathbf{u}^N,\vartheta)]}{\partial \vartheta^2}] \in R^{\bar{D}\times\bar{D}}.$$

When AIC is applied, it is assumed $J(\vartheta) \approx I(\vartheta)$ [74], i.e., only good approximation models will be compared in terms of the AIC where "good" means these models $\bar{f}$ are quite near the true model $\bar{f}_0$ in terms of the KL distance. Thus, the following approximation holds

$$\begin{aligned}\mathbb{E}_{z^N}[\mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))]]] &\approx \mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(y^N))]] - \mathbf{tr}[\mathbf{I}_{\bar{D}}] \\ &= \mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(y^N))]] - \bar{D}\end{aligned} \tag{3.30}$$

where $\mathbf{I}_{\bar{D}} \in R^{\bar{D}\times\bar{D}}$ is an identity matrix. AIC is then defined as

$$\text{AIC} \triangleq -2\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(y^N))] + 2\bar{D} \tag{3.31}$$

which is an estimate of $-2\cdot\mathbb{E}_{z^N}[\mathbb{E}_{y^N}[\log[\bar{f}(y^N|\mathbf{u}^N,\hat{\vartheta}(z^N))]]]$. Combining Eq. (3.25) and Eq. (3.31) gives

$$\begin{aligned}\text{AIC} =& -2\log[\prod_{t=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}\cdot\exp[-\frac{1}{2\sigma^2}(y(t)-\hat{y}(t|t-1))^2]] + 2\bar{D} \\ =& N\log(2\pi) + N\log(\sigma^2) + \frac{1}{\sigma^2}\sum_{t=1}^{N}|\epsilon(t)|^2 + 2\bar{D}\end{aligned} \tag{3.32}$$

where the parameter σ^2 is estimated by

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{t=1}^{N}|\epsilon(t)|^2. \tag{3.33}$$

Thus, by omitting those constant terms in Eq. (3.32), we have

$$\text{AIC} = N\log(\hat{\sigma}^2) + 2\bar{D} \tag{3.34}$$

The most appropriate model should have the minimum AIC value. From Eq. (3.34), a good model should have not only small prediction errors indicated by $N\log(\hat{\sigma}^2)$ but a

small dimension $\bar{D}$ for ϑ. Furthermore, if the model structure is some linear regression prediction model with the corresponding prediction error sufficing Normal$(0, \sigma^2)$, the small sample AIC, or specifically, Akaike's information criterion with a correction for finite sample sizes (AIC$_c$), can be used for model order selection. The benefit is that when the sample size N is small, AIC$_c$ gives additional correction term which penalises performances of any model with a large dimension for ϑ. According to [74], AIC$_c$ is given as follows:

$$\text{AIC}_c = \text{AIC} + \frac{2\bar{D}(\bar{D}+1)}{N-\bar{D}-1} \tag{3.35}$$

where $2\bar{D}(\bar{D}+1)/(t-\bar{D}-1)$ is the correction term. Note that Eq. (3.31) is a general form which can be used to select models for any assumed f_e while Eq. (3.34) and Eq. (3.35) can be explicitly derived only if the prediction error, $\epsilon(t)$, is normally distributed. Unfortunately, there exists no unique correction term for AIC$_c$ to deal with any type of assumed model structures. However, Eq. (3.35) may be still used when there is no other better correction method [74].

3.2.5 Model validation

The residual analysis [60] suggests a validated model has residuals $\epsilon(t)$ which are serially independent and unrelated to past inputs. Two correlation-based intermediate variables are defined as

$$\hat{R}_{\epsilon}^{N}(\tau) = (1/N)\sum_{t=1}^{N}\epsilon(t)\epsilon(t-\tau) \tag{3.36}$$

and

$$\hat{R}_{\epsilon u}^{N}(\tau) = (1/N)\sum_{t=1}^{N}\epsilon(t)u(t-\tau). \tag{3.37}$$

$\zeta_1(\tau)$ and $\zeta_2(\tau)$ are then defined as

$$\zeta_1(\tau) = (N/\hat{\sigma}^4)\cdot(\hat{R}_{\epsilon}^{N}(\tau))^2 \sim \chi^2(1) \tag{3.38}$$

and

$$\zeta_2(\tau) = \sqrt{N/(\hat{\sigma}^2 P(\tau))}\hat{R}_{\epsilon u}^{N}(\tau) \sim \mathcal{N}(0,\,1) \tag{3.39}$$

with $P(\tau) = (1/N)\sum_{t=1}^{N} u(t-\tau)^2$. For a validated model, $\zeta_1(\tau)$ and $\zeta_2(\tau)$ should be within the α-level confidence intervals determined by the chi-squared- and normally-distributed random variables, respectively.

3.3 Generalised predictive control

3.3.1 Preliminaries

In this section, a popular MPC algorithm called GPC [79, 80] is introduced and applied to a validated neural-network-based carbon capture plant model. One intriguing characteristic of GPC is that it can be applied with only measured input-output data from a plant. However, this control protocol is model-based, i.e., a prediciton model of the controlled plant is necessary for the control implementation. Consider the prediciton model

$$\mathbf{A}(q)y(t) = \mathbf{B}(q)u(t-1) + \mathbf{D}(q)v(t) + \frac{\mathbf{C}(q)e(t)}{\Delta} \tag{3.40}$$

where $y \in \mathbb{R}^{\bar{n}}$, $u \in \mathbb{R}^{\bar{m}}$, $v \in \mathbb{R}^{\bar{p}}$, and $\Delta = 1 - q^{-1}$ which is a transfer operator. Besides Δ, the transfer operator matrices of the prediction model are represented by

$$\begin{cases} \mathbf{A}(q) \triangleq I + A_1 q^{-1} + A_2 q^{-1} + \cdots + A_{n_a} q^{-n_a} \\ \mathbf{B}(q) \triangleq B_0 + B_1 q^{-1} + B_2 q^{-1} + \cdots + B_{n_b-1} q^{-n_b+1} \\ \mathbf{C}(q) \triangleq I + C_1 q^{-1} + C_2 q^{-1} + \cdots + C_{n_c-1} q^{-n_c} \\ \mathbf{D}(q) \triangleq D_0 + D_1 q^{-1} + D_2 q^{-2} + \cdots + D_{n_d-1} q^{-n_d+1} \end{cases} \tag{3.41}$$

where $A_i \in \mathbb{R}^{\bar{n}\times\bar{n}}$, $B_i \in \mathbb{R}^{\bar{n}\times\bar{m}}$, $C_i \in \mathbb{R}^{\bar{n}\times\bar{n}}$ and $D_i \in \mathbb{R}^{\bar{n}\times\bar{p}}$. With the weight matrices $\mathbf{Q} \in \mathbb{R}^{N_r\bar{n}\times N_r\bar{n}}$, $\mathbf{R} \in \mathbb{R}^{N_r\bar{m}\times N_r\bar{m}}$, and the definitions of

$$\begin{cases} \bar{\mathbf{U}}(t) \triangleq (u(t), u(t+1), \cdots, u(t+N_r-1))^T \\ \bar{\mathbf{Y}}(t) \triangleq (\hat{y}(t+1|t), \hat{y}(t+2|t), \cdots, \hat{y}(t+N_r|t))^T \\ \bar{\mathbf{Y}}_r(t) \triangleq (y_r(t+1), y_r(t+2), \cdots, y_r(t+N_r))^T \end{cases} \tag{3.42}$$

where N_r is the terminal time index of the output prediction $\bar{\mathbf{Y}}(t)$, the objective of the GPC protocol can be defined as

$$J = (\bar{\mathbf{Y}}_r(t) - \bar{\mathbf{Y}}(t))^T \mathbf{Q} (\bar{\mathbf{Y}}_r(t) - \bar{\mathbf{Y}}(t)) + \Delta\bar{\mathbf{U}}^T(t)\mathbf{R}\Delta\bar{\mathbf{U}}(t). \tag{3.43}$$

It follows that the optimal GPC protocol can be formulated by solving the problem of

$$\Delta\bar{\mathbf{U}}^*(t) = \arg\min_{\Delta\bar{\mathbf{U}}(t)} J. \tag{3.44}$$

A necessary condition for the optimal control protocol (Eq. (3.44)) is

$$\frac{\partial J}{\partial \Delta\bar{\mathbf{U}}(t)} = 0. \tag{3.45}$$

For the application of GPC in the real system, a stable and feasible solution must also consider the constraints of the inputs and states [25]. Nonetheless, the main purpose of the thesis is to compare the performances of GPC with the MFAC in Chapter 4. For simplicity, we only consider the unconstrained cases. Robust MPC may cause conservative input manipulations which degrade the performances [81].

3.3.2 Control algorithm

Note that we actually consider a carbon capture process where the lean MEA flow rate $u \in \mathbb{R}$ and carbon capture level $y \in \mathbb{R}$ are the manipulated input and controlled output, respectively. In addition, there exist disturbances, denoted by $v \triangleq (v_1, v_2)^T \in \mathbb{R}^2$, coming from two channels, i.e., the flue gas flow rate v_1 and the mass fraction of CO_2 in the flue gas v_2. Thus, the GPC protocol of the carbon capture process should be formulated in terms of the single-input-single-output framework with two disturbance channels. It infers that a prediciton model of the controlled plant should be specified as Eq. (3.40) with the corresponding transfer operator matrices (Eq. (3.41)) reduced to the scalar cases, i.e.,

$$\begin{cases} \mathbf{A}(q) \triangleq 1 + a_1 q^{-1} + a_2 q^{-1} + \cdots + a_{n_a} q^{-n_a} \\ \mathbf{B}(q) \triangleq b_0 + b_1 q^{-1} + b_2 q^{-1} + \cdots + b_{n_b-1} q^{-n_b+1} \\ \mathbf{C}(q) \triangleq 1 + c_1 q^{-1} + c_2 q^{-1} + \cdots + c_{n_c-1} q^{-n_c} \\ \mathbf{D}(q) \triangleq \begin{pmatrix} \mathbf{D}^1(q) \\ \mathbf{D}^2(q) \end{pmatrix}^T \triangleq \begin{pmatrix} d_0^1 + d_1^1 q^{-1} + d_2^1 q^{-1} + \cdots + d_{n_{d1}}^1 q^{-n_{d1}} \\ d_0^2 + d_1^2 q^{-1} + d_2^2 q^{-1} + \cdots + d_{n_{d2}}^2 q^{-n_{d2}} \end{pmatrix}^T \end{cases} \tag{3.46}$$

where Eq. (3.46) indicates the more general representation of $\mathbf{D}(q)$ in Eq. (3.46) should have $n_d = \max\{n_{d1}, n_{d2}\} + 1$. In addition, the preceding neural-network-based model

structure implies $d_0^1 = d_0^2 = 0$ since there exist explicit one-sample delay between the inputs and outputs for any model in the model set (Eq. (3.14)). With the weight matrices $\mathbf{Q} \in \mathbb{R}^{N_r \times N_r}$, $\mathbf{R} \in \mathbb{R}^{N_r \times N_r}$, and the definitions of

$$\begin{cases} \bar{\mathbf{U}}(t) \triangleq (u(t), u(t+1), \cdots, u(t+N_r-1))^T \\ \bar{\mathbf{V}}(t) \triangleq (v^T(t+1), v^T(t+2), \cdots, v^T(t+N_r))^T \\ \bar{\mathbf{Y}}(t) \triangleq (\hat{y}(t+1|t),\ \hat{y}(t+2|t), \cdots, \hat{y}(t+N_r|t))^T \\ \bar{\mathbf{Y}}_r(t) \triangleq (y_r(t+1),\ y_r(t+2), \cdots, y_r(t+N_r))^T \end{cases} \tag{3.47}$$

the objective of the GPC protocol is exactly identical to Eq. (3.43). In the following parts of this subsection, the GPC is formulated by solving the problem of Eq. (3.44).

k-step-ahead prediction

First, we introduce the k-step-ahead prediciton which is essential to obtain $\bar{\mathbf{Y}}(t)$. Note that the prediciton model (Eq. (3.40)) can be rewritten as

$$y(t+k) = \frac{\mathbf{B}(q)}{\mathbf{A}(q)} u(t+k-1) + \frac{\mathbf{D}(q)}{\mathbf{A}(q)} v(t+k) + \frac{\mathbf{C}(q)}{\mathbf{A}(q)\Delta} e(t+k). \tag{3.48}$$

To achieve the k-step-ahead prediciton, the coloured noise term $\mathbf{C}(q)/(\mathbf{A}(q)\Delta)e(t+k)$ should be divided into two parts: One part is due to the noises of the past time, i.e., $e_{-\infty}^{t} = \{e(-\infty), \cdots, e(t-1), e(t)\}$, while the other part is due to the noises of the future time, i.e., $e_{t+1}^{t+k} = \{e(t+1), e(t+2), ..., e(t+k)\}$. Equivalently, we should obtain

$$\frac{\mathbf{C}(q)}{\mathbf{A}(q)\Delta} = \mathbf{E}_k(q) + q^{-k}\tilde{\mathbf{F}}_k(q) \tag{3.49}$$

where

$$\begin{cases} \mathbf{E}_k(q) \triangleq & e_{k,0} + e_{k,1}q^{-1} + e_{k,2}q^{-2} + \cdots + e_{k,k-1}q^{-k+1} \\ \tilde{\mathbf{F}}_k(q) \triangleq & \tilde{f}_{k,0} + \tilde{f}_{k,1}q^{-1} + \tilde{f}_{k,2}q^{-2} + \cdots + \tilde{f}_{k,\infty}q^{-\infty} \end{cases} \tag{3.50}$$

to achieve such a division. Eq. (3.49) implies the well-known Diophantine equation

$$\mathbf{C}(q) = \mathbf{E}_k(q)\mathbf{A}(q)\Delta + q^{-k}\mathbf{F}_k(q) \tag{3.51}$$

where

$$\begin{aligned} \mathbf{F}_k(q) \triangleq & \tilde{\mathbf{F}}_k(q)\mathbf{A}(q)\Delta \\ = & f_{k,0} + f_{k,1}q^{-1} + f_{k,2}q^{-2} + \cdots + f_{k,n_a}q^{-n_a}. \end{aligned} \tag{3.52}$$

Denoting $\delta(\cdot)$ as the degree of a transfer operator, it can be derived that $\delta(\mathbf{C}(q)) = n_c$ and $\delta(\mathbf{E}_k(q)\mathbf{A}(q)\Delta) = n_a+k$. With the assumption $n_a+k \geq n_c$, it infers $\delta(\mathbf{F}_k(q)) = n_a$ as in Eq. (3.52). Supposed that the transfer operators, $\mathbf{E}_k(q)$ and $\mathbf{F}_k(q)$, are known already, multiplying Eq. (3.48) by $\mathbf{E}_k(q)\mathbf{A}(q)\Delta$ derives

$$\begin{aligned}\mathbf{E}_k(q)\mathbf{A}(q)\Delta y(t+k) =&\mathbf{E}_k(q)\mathbf{B}(q)\Delta u(t+k-1)\\ &+\mathbf{E}_k(q)\mathbf{D}(q)\Delta v(t)+\mathbf{E}_k(q)\mathbf{C}(q)e(t+k).\end{aligned}\tag{3.53}$$

Combining Eq. (3.51) and Eq. (3.53) gives

$$\begin{aligned}\mathbf{C}(q)y(t+k)-\mathbf{F}_k(q)y(t) =&\mathbf{E}_k(q)\mathbf{B}(q)\Delta u(t+k-1)\\ &+\mathbf{E}_k(q)\mathbf{D}(q)\Delta v(t+k)+\mathbf{E}_k(q)\mathbf{C}(q)e(t+k)\end{aligned}\tag{3.54}$$

which indicates

$$\begin{aligned}y(t+k)-\frac{\mathbf{F}_k(q)}{\mathbf{C}(q)}y(t) =&\frac{\mathbf{E}_k(q)\mathbf{B}(q)}{\mathbf{C}(q)}\Delta u(t+k-1)\\ &+\frac{\mathbf{E}_k(q)\mathbf{D}(q)}{\mathbf{C}(q)}\Delta v(t+k)+\mathbf{E}_k(q)e(t+k)\end{aligned}\tag{3.55}$$

where $\mathbf{E}_k(q)e(t+k)$ only involves the noises after time t. Therefore,

$$\mathbf{C}(q)\hat{y}(t+k|t)-\mathbf{F}_k(q)y(t) =\mathbf{E}_k(q)\mathbf{B}(q)\Delta u(t+k-1)+\mathbf{E}_k(q)\mathbf{D}(q)\Delta v(t+k).\tag{3.56}$$

Constructing another Diophantine equation

$$1=\mathbf{M}_k(q)\mathbf{C}(q)+q^{-k}\mathbf{N}_k(q)\tag{3.57}$$

where

$$\begin{cases}\mathbf{M}_k(q)\triangleq & m_{k,0}+m_{k,1}q^{-1}+m_{k,2}q^{-2}+\cdots+m_{k,k-1}q^{-k+1}\\ \mathbf{N}_k(q)\triangleq & n_{k,0}+n_{k,1}q^{-1}+n_{k,2}q^{-2}+\cdots+n_{k,n_c-1}q^{-n_c+1}\end{cases}\tag{3.58}$$

and multiplying Eq. (3.56) by $\mathbf{M}_k(q)$ derives

$$\begin{aligned}\hat{y}(t+k|t) =&\mathbf{M}_k(q)\mathbf{E}_k(q)\mathbf{B}(q)\Delta u(t+k-1)+\mathbf{M}_k(q)\mathbf{E}_k(q)\mathbf{D}(q)\Delta v(t+k)\\ &+(\mathbf{M}_k(q)\mathbf{F}_k(q)+\mathbf{N}_k(q))y(t)\end{aligned}\tag{3.59}$$

which is a k-step-ahead prediciton of $y(t+k)$.

Note that a GPC protocol will be applied to such a predictor model (Eq. (3.59)). The input should be categorised into the future and past parts where the future inputs

can be manipulated in terms of some control protocol to achieve a specified objective while the past inputs contribute to some free response of the plant. Suppose that

$$\mathbf{M}_k(q)\mathbf{E}_k(q)\mathbf{B}(q) = \mathbf{G}_{\mathrm{f}k}(q) + q^{-k}\mathbf{G}_{\mathrm{p}k}(q) \tag{3.60}$$

$$\mathbf{M}_k(q)\mathbf{E}_k(q)\mathbf{D}(q) = \mathbf{H}_{\mathrm{f}k}(q) + q^{-k}\mathbf{H}_{\mathrm{p}k}(q) \tag{3.61}$$

subject to

$$\begin{cases} \mathbf{G}_{\mathrm{f}k}(q) \triangleq g_{\mathrm{f}k,0} + g_{\mathrm{f}k,1}q^{-1} + g_{\mathrm{f}k,2}q^{-2} + \cdots + g_{\mathrm{f}k,k-1}q^{-k+1} \\ \mathbf{G}_{\mathrm{p}k}(q) \triangleq g_{\mathrm{p}k,0} + g_{\mathrm{p}k,1}q^{-1} + g_{\mathrm{p}k,2}q^{-2} + \cdots + g_{\mathrm{p}k,k+n_b-3}q^{-k-n_b+3} \\ \mathbf{H}_{\mathrm{f}k}(q) \triangleq \begin{pmatrix} \mathbf{H}^1_{\mathrm{f}k}(q) \\ \mathbf{H}^2_{\mathrm{f}k}(q) \end{pmatrix}^T \triangleq \begin{pmatrix} h^1_{\mathrm{f}k,0} + h^1_{\mathrm{f}k,1}q^{-1} + h^1_{\mathrm{f}k,2}q^{-2} + \cdots + h^1_{\mathrm{f}k,k-1}q^{-k+1} \\ h^2_{\mathrm{f}k,0} + h^2_{\mathrm{f}k,1}q^{-1} + h^2_{\mathrm{f}k,2}q^{-2} + \cdots + h^2_{\mathrm{f}k,k-1}q^{-k+1} \end{pmatrix}^T \\ \mathbf{H}_{\mathrm{p}k}(q) \triangleq \begin{pmatrix} \mathbf{H}^1_{\mathrm{p}k}(q) \\ \mathbf{H}^2_{\mathrm{p}k}(q) \end{pmatrix}^T \triangleq \begin{pmatrix} h^1_{\mathrm{p}k,0} + h^1_{\mathrm{p}k,1}q^{-1} + h^1_{\mathrm{p}k,2}q^{-2} + \cdots + h^1_{\mathrm{p}k,k+n_{d1}-2}q^{-k-n_{d1}+2} \\ h^2_{\mathrm{p}k,0} + h^2_{\mathrm{p}k,1}q^{-1} + h^2_{\mathrm{p}k,2}q^{-2} + \cdots + h^2_{\mathrm{p}k,k+n_{d2}-2}q^{-k-n_{d2}+2} \end{pmatrix}^T \end{cases} \tag{3.62}$$

where the element-wise degrees of $\mathbf{G}_{\mathrm{p}k}(q)$ and $\mathbf{H}_{\mathrm{p}k}(q)$ are

$$\begin{cases} \delta(\mathbf{G}_{\mathrm{p}k}(q)) = & \delta(\mathbf{M}_k(q)\mathbf{E}_k(q)\mathbf{B}(q)) - k = k + n_b - 3 \\ \delta(\mathbf{H}_{\mathrm{p}k}(q)) = & \delta(\mathbf{M}_k(q)\mathbf{E}_k(q)\mathbf{D}(q)) - k \\ & = \begin{pmatrix} k + n_{d1} - 2 \\ k + n_{d2} - 2 \end{pmatrix} \end{cases} . \tag{3.63}$$

Eq.(3.59) can be rewritten as

$$\begin{aligned} \hat{y}(t+k|t) =& \mathbf{G}_{\mathrm{f}k}(q)\Delta u(t+k-1) + \mathbf{G}_{\mathrm{p}k}(q)\Delta u(t-1) \\ &+ \mathbf{H}_{\mathrm{f}k}(q)\Delta v(t+k) + \mathbf{H}_{\mathrm{p}k}(q)\Delta v(t) + (\mathbf{M}_k(q)\mathbf{F}_k(q) + \mathbf{N}_k(q))y(t) \end{aligned} \tag{3.64}$$

which is a k-step-ahead predictor model. Note that the transfer operators such as $\mathbf{G}_{\mathrm{f}k}(q)$, $\mathbf{G}_{\mathrm{p}k}(q)$, and etc., in this predictor model can be obtained when the aforementioned Diophantine equations, Eq. (3.51) and Eq. (3.57), are solved. The next part will discuss a simple method to solving these equations recursively.

Recursion of Diophantine equations

As mentioned above, to know the exact representation of a k-step-ahead prediciton (Eq. (3.64)), the transfer operators $\mathbf{E}_k(q)$, $\mathbf{F}_k(q)$, $\mathbf{M}_k(q)$ and $\mathbf{N}_k(q)$ for $\forall k \in \{1, 2, \cdots, N_r\}$ in Diophantine equations (Eqs. (3.51) and (3.57)) should be obtained.

For Eq. (3.51), when $k = 1$, it can be derived that

$$\mathbf{C}(q) = \mathbf{E}_1(q)\tilde{\mathbf{A}}(q) + q^{-1}\mathbf{F}_1(q) \tag{3.65}$$

which gives

$$\mathbf{E}_1(q) = c_0 \tag{3.66}$$

$$\mathbf{F}_1(q) = q(\mathbf{C}(q) - c_0\tilde{\mathbf{A}}(q)). \tag{3.67}$$

When $k > 1$, Eq. (3.51) indicates

$$\mathbf{C}(q) = \mathbf{E}_{k-1}(q)\mathbf{A}(q)\Delta + q^{-k+1}\mathbf{F}_{k-1}(q). \tag{3.68}$$

Subtracting Eq. (3.51) for both sides of Eq. (3.68) gives

$$\begin{aligned} 0 =& (\mathbf{E}_k(q) - \mathbf{E}_{k-1}(q))\mathbf{A}(q)\Delta + q^{-k+1}(q^{-1}\mathbf{F}_k(q) - \mathbf{F}_{k-1}(q)) \\ =& q^{-k+1}(e_{k,k-1}\mathbf{A}(q)\Delta + q^{-1}\mathbf{F}_k(q) - \mathbf{F}_{k-1}(q)). \end{aligned} \tag{3.69}$$

By denoting

$$\tilde{\mathbf{A}}(q) = \mathbf{A}(q)\Delta = 1 + \tilde{a}_1 q^{-1} + \tilde{a}_2 q^{-2} + \cdots + \tilde{a}_{n_a+1} q^{-n_a-1} \tag{3.70}$$

together with Eq. (3.69), it can be derived that

$$\begin{cases} 0 = e_{k,\nu} - e_{k-1,\nu},\ \nu \in \{0, 1, \cdots, k-2\} \\ 0 = e_{k,\nu} - f_{k-1,0},\ \nu \in \{k-1\} \\ 0 = e_{k,k-1}\tilde{a}_{l+1} + f_{k,l} - f_{k-1,l+1},\ l \in \{0, 1, \cdots, n_a - 1\} \\ 0 = e_{k,k-1}\tilde{a}_{l+1} + f_{k,l},\ l \in \{n_a\} \end{cases} \tag{3.71}$$

i.e.,

$$\begin{cases} e_{k,\nu} = e_{k-1,\nu},\ \nu \in \{0, 1, \cdots, k-2\} \\ e_{k,\nu} = f_{k-1,0},\ \nu \in \{k-1\} \\ f_{k,l} = -e_{k,k-1}\tilde{a}_{l+1} + f_{k-1,l+1},\ l \in \{0, 1, \cdots, n_a - 1\} \\ f_{k,l} = -e_{k,k-1}\tilde{a}_{l+1},\ l \in \{n_a\} \end{cases} . \tag{3.72}$$

Thus, a user can recursively obtain $\mathbf{E}_k(q)$ and $\mathbf{F}_k(q)$ with the above four equations for $k \in \{2, \cdots, N_r\}$ with the initial conditions of $\mathbf{E}_1(q)$ (Eq. (3.66)) and $\mathbf{F}_1(q)$ (Eq. (3.67)).

A similar recursive method can be applied to the other Diophantine equation (Eq. (3.57)) where the initial conditions can be expressed as

$$\mathbf{M}_1(q) = 1/c_0 \tag{3.73}$$

$$\mathbf{N}_1(q) = q(1 - c_0^{-1}\mathbf{C}(q)). \tag{3.74}$$

In addition, since

$$\begin{aligned} 0 =& (\mathbf{M}_k(q) - \mathbf{M}_{k-1}(q))\mathbf{C}(q) + q^{-k+1}(q^{-1}\mathbf{N}_k(q) - \mathbf{N}_{k-1}(q)) \\ =& q^{-k+1}(m_{k,k-1}\mathbf{C}(q) + q^{-1}\mathbf{N}_k(q) - \mathbf{N}_{k-1}(q)) \end{aligned} \tag{3.75}$$

holds, it implies

$$\begin{cases} m_{k,\nu} = m_{k-1,\nu},\ \nu \in \{0, 1, \cdots, k-2\} \\ m_{k,\nu} = n_{k-1,0}/c_0,\ \nu \in \{k-1\} \\ n_{k,l} = -m_{k,k-1}c_{l+1} + n_{k-1,l+1},\ l \in \{0, 1, \cdots, n_c - 2\} \\ n_{k,l} = -m_{k,k-1}c_{l+1},\ l \in \{n_c - 1\} \end{cases}. \tag{3.76}$$

One can now recursively obtain $\mathbf{M}_k(q)$ and $\mathbf{N}_k(q)$ with the above four equations for $k \in \{2, \cdots, N_r\}$ with the initial conditions of $\mathbf{M}_1(q)$ (Eq. (3.73)) and $\mathbf{N}_1(q)$ (Eq. (3.74)).

Control protocol

The control protocol is then derived as follows. Denoting

$$\begin{cases} \mathbf{E}_{k+1}(q) = \tilde{\mathbf{E}}_{k+1}(q) + e_{k+1,k}q^{-k} \\ \mathbf{M}_{k+1}(q) = \tilde{\mathbf{M}}_{k+1}(q) + m_{k+1,k}q^{-k} \end{cases} \tag{3.77}$$

it can be inferred that

$$\begin{cases} \tilde{\mathbf{E}}_{k+1}(q) = \mathbf{E}_k(q) \\ \tilde{\mathbf{M}}_{k+1}(q) = \mathbf{M}_k(q) \end{cases} \tag{3.78}$$

from the first equations of both Diophantine recursion methods (Eqs. (3.72) and (3.76)). Thus, defining

$$\mathbf{G}_{\mathrm{f}k+1}(q) = \tilde{\mathbf{G}}_{\mathrm{f}k+1}(q) + q^{-k} g_{\mathrm{f}k+1,k} \tag{3.79}$$

$$\mathbf{H}_{\mathrm{f}k+1}(q) = \tilde{\mathbf{H}}_{\mathrm{f}k+1}(q) + q^{-k} \begin{pmatrix} h^1_{\mathrm{f}k+1,k} \\ h^2_{\mathrm{f}k+1,k} \end{pmatrix}^T \tag{3.80}$$

together with Eq. (3.60) and Eq. (3.61) gives

$$\begin{cases} \tilde{\mathbf{G}}_{\mathrm{f}k+1}(q) = \mathbf{G}_{\mathrm{f}k}(q) \\ \tilde{\mathbf{H}}_{\mathrm{f}k+1}(q) = \mathbf{H}_{\mathrm{f}k}(q) \end{cases} . \tag{3.81}$$

From Eq. (3.81), we have

$$\begin{cases} g_{\mathrm{f}k,k-1} = g_{\mathrm{f}k+1,k-1} = \cdots = g_{\mathrm{f}N_r,k-1} \triangleq g_{k-1} \\ h^1_{\mathrm{f}k,k-1} = h^1_{\mathrm{f}k+1,k-1} = \cdots = h^1_{\mathrm{f}N_r,k-1} \triangleq h^1_{k-1} \\ h^2_{\mathrm{f}k,k-1} = h^2_{\mathrm{f}k+1,k-1} = \cdots = h^2_{\mathrm{f}N_r,k-1} \triangleq h^2_{k-1} \end{cases} \tag{3.82}$$

where $k \in \{1, 2, \cdots, N_r\}$. Therefore, in terms of Eq. (3.64) and Eq. (3.82), the concerned output prediction vector $(\hat{y}(t+1|t),\ \hat{y}(t+2|t),\ \cdots,\ \hat{y}(t+N_r|t))^T$ can be represented as

$$\begin{aligned} \bar{\mathbf{Y}}(t) \triangleq \begin{pmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \hat{y}(t+3|t) \\ \vdots \\ \hat{y}(t+N_r|t) \end{pmatrix} &= \begin{pmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_r-1} & g_{N_r-2} & g_{N_r-3} & \cdots & g_0 \end{pmatrix} \begin{pmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \\ \vdots \\ \Delta u(t+N_r-1) \end{pmatrix} \\ &+ \begin{pmatrix} h^1_0 & 0 & 0 & \cdots & 0 & h^2_0 & 0 & 0 & \cdots & 0 \\ h^1_1 & h^1_0 & 0 & \cdots & 0 & h^2_1 & h^2_0 & 0 & \cdots & 0 \\ h^1_2 & h^1_1 & h^1_0 & \cdots & 0 & h^2_2 & h^2_1 & h^2_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h^1_{N_r-1} & h^1_{N_r-2} & h^1_{N_r-3} & \cdots & h^1_0 & h^2_{N_r-1} & h^2_{N_r-2} & h^2_{N_r-3} & \cdots & h^2_0 \end{pmatrix} \\ &\cdot \Big(\Delta v_1(t+1) \quad \Delta v_1(t+2) \quad \cdots \quad \Delta v_1(t+N_r) \\ &\qquad \Delta v_2(t+1) \quad \Delta v_2(t+2) \quad \cdots \quad \Delta v_2(t+N_r) \Big)^T \end{aligned}$$

$$+\begin{pmatrix} \mathbf{G}_{\mathrm{p}1}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}1}(q)\Delta v(t)+(\mathbf{M}_1(q)\mathbf{F}_1(q)+\mathbf{N}_1(q))y(t) \\ \mathbf{G}_{\mathrm{p}2}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}2}(q)\Delta v(t)+(\mathbf{M}_2(q)\mathbf{F}_2(q)+\mathbf{N}_2(q))y(t) \\ \mathbf{G}_{\mathrm{p}3}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}3}(q)\Delta v(t)+(\mathbf{M}_3(q)\mathbf{F}_3(q)+\mathbf{N}_3(q))y(t) \\ \vdots \\ \mathbf{G}_{\mathrm{p}N_r}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}N_r}(q)\Delta v(t)+(\mathbf{M}_{N_r}(q)\mathbf{F}_{N_r}(q)+\mathbf{N}_{N_r}(q))y(t) \end{pmatrix}. \quad (3.83)$$

Denoting

$$\mathbf{G} \triangleq \begin{pmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_r-1} & g_{N_r-2} & g_{N_r-3} & \cdots & g_0 \end{pmatrix} \quad (3.84)$$

$$\mathbf{H} \triangleq \begin{pmatrix} h_0^1 & 0 & 0 & \cdots & 0 & h_0^2 & 0 & 0 & \cdots & 0 \\ h_1^1 & h_0^1 & 0 & \cdots & 0 & h_1^2 & h_0^2 & 0 & \cdots & 0 \\ h_2^1 & h_1^1 & h_0^1 & \cdots & 0 & h_2^2 & h_1^2 & h_0^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ h_{N_r-1}^1 & h_{N_r-2}^1 & h_{N_r-3}^1 & \cdots & h_0^1 & h_{N_r-1}^2 & h_{N_r-2}^2 & h_{N_r-3}^2 & \cdots & h_0^2 \end{pmatrix} \quad (3.85)$$

$$\mathbf{f}(t) \triangleq \begin{pmatrix} \mathbf{G}_{\mathrm{p}1}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}1}(q)\Delta v(t)+(\mathbf{M}_1(q)\mathbf{F}_1(q)+\mathbf{N}_1(q))y(t) \\ \mathbf{G}_{\mathrm{p}2}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}2}(q)\Delta v(t)+(\mathbf{M}_2(q)\mathbf{F}_2(q)+\mathbf{N}_2(q))y(t) \\ \mathbf{G}_{\mathrm{p}3}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}3}(q)\Delta v(t)+(\mathbf{M}_3(q)\mathbf{F}_3(q)+\mathbf{N}_3(q))y(t) \\ \vdots \\ \mathbf{G}_{\mathrm{p}N_r}(q)\Delta u(t-1)+\mathbf{H}_{\mathrm{p}N_r}(q)\Delta v(t)+(\mathbf{M}_{N_r}(q)\mathbf{F}_{N_r}(q)+\mathbf{N}_{N_r}(q))y(t) \end{pmatrix} \quad (3.86)$$

$$\begin{aligned} \tilde{\mathbf{V}}(t) \triangleq (&v_1(t+1), v_1(t+2), \cdots, v_1(t+N_r), \\ &v_2(t+1), v_2(t+2), \cdots, v_2(t+N_r))^T \end{aligned} \quad (3.87)$$

Eq. (3.83) can be briefly written as

$$\bar{\mathbf{Y}}(t) = \mathbf{G}\Delta\bar{\mathbf{U}}(t) + \mathbf{H}\Delta\tilde{\mathbf{V}}(t) + \mathbf{f}(t). \quad (3.88)$$

Note that $\tilde{\mathbf{V}}(t)$ can be formulated by $\bar{\mathbf{V}}(t)$ through a permutation matrix, and the second term to the right-hand side of Eq. (3.88), $\mathbf{H}\Delta\tilde{\mathbf{V}}(t)$, is the response of future disturbances which can be obtained if they are measurable. Define

$$\mathbf{f}'(t) \triangleq \mathbf{H}\Delta\tilde{\mathbf{V}}(t) + \mathbf{f}(t). \tag{3.89}$$

Combining Eq. (3.88), Eq. (3.89), and the objective function (Eq. (3.43)) gives

$$J = (\mathbf{G}\Delta\bar{\mathbf{U}}(t) + \mathbf{f}'(t) - \bar{\mathbf{Y}}_r(t))^T\mathbf{Q}(\mathbf{G}\Delta\bar{\mathbf{U}}(t) + \mathbf{f}'(t) - \bar{\mathbf{Y}}_r(t)) + \Delta\bar{\mathbf{U}}^T(t)\mathbf{R}\Delta\bar{\mathbf{U}}(t) \tag{3.90}$$

where $\mathbf{f}'(t)$ is a filtered response [25]. Taking $\partial J/\partial\bar{\mathbf{U}}(t) = 0$, the control policy is then derived as

$$\Delta\bar{\mathbf{U}}^*(t) = (\mathbf{G}^T\mathbf{Q}\mathbf{G} + \mathbf{R})^{-1}\mathbf{G}^T\mathbf{Q}(\bar{\mathbf{Y}}_r(t) - \mathbf{f}'(t)) \tag{3.91}$$

where only the first element of the vector $\Delta\bar{\mathbf{U}}^*(t)$ is implemented for the controlled plant. Note that for a model-based protocol, the underline model parameters from sensitivity or identification tests are usually required. For this specific GPC algorithm, the model parameters are coefficients of the transfer operators $\mathbf{A}(q)$, $\mathbf{B}(q)$ and $\mathbf{D}(q)$ which can approximate the PCC plant in some standard mathematical form (Eq. (3.40)). These model parameters are the indispensable priori knowledge for the model-based control design. To implement the control protocol (Eq. (3.91)), both the matrices $\mathbf{G}$ & $\mathbf{H}$ and the filter $\mathbf{f}'(t)$ should be determined by $\mathbf{A}(q)$, $\mathbf{B}(q)$ and $\mathbf{D}(q)$ as above, which infers that GPC should be model-based.

3.3.3 Numerical example

In this subsection, three GPC controllers with different prediciton models are designed and applied to a plant which can be described by

$$(1 - 0.8q^{-1})y(t) = (0.4 + 0.6q^{-1})u(t-1) + (0.7q^{-1} + 0.3q^{-2})v(t) + \frac{1}{\Delta}e(t). \tag{3.92}$$

The tuning parameters N_r, $\mathbf{Q}$ and $\mathbf{R}$ of all the three controllers are set to be identical, i.e., $N_r = 3$ and $\mathbf{Q} = \mathbf{R} = I_{3\times3}$. Nevertheless, some differences are given

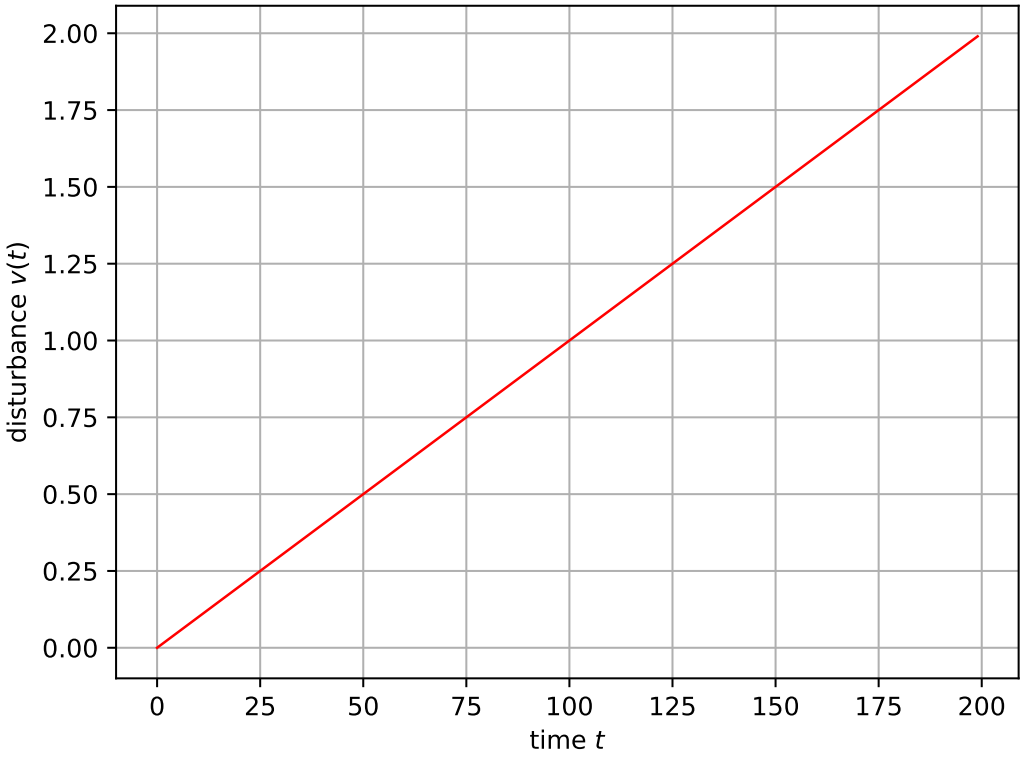


Figure 3.3: Numerical example: Disturbance of a plant with GPC.

as follows: The first controller has no disturbance model, i.e., $\mathbf{D}(q) = 0$; the second controller has a disturbance model, $\mathbf{D}(q) = 0.7q^{-1} + 0.3q^{-2}$ but is uncertain about the future disturbance; the third controller has the disturbance model and knows the future disturbance deviations. Some other transfer operators when applicable to these controllers are exactly identical to the plant specification (Eq. (3.92)), which implies, $\mathbf{A}(q) = 1 - 0.8q^{-1}$, $\mathbf{B}(q) = 0.4 + 0.6q^{-1}$, and $\mathbf{C}(q) = 1$. The assumed disturbance and reference signal are shown in Figure 3.3 and Figure 3.5, respectively, while the manipulated input and controlled output for the three controllers are displayed in Figure 3.4 and Figure 3.5, respectively. On the ground of the tacking performances (Figure 3.5), the GPC protocol with both disturbance model and disturbance prediction is the best among the three controllers.

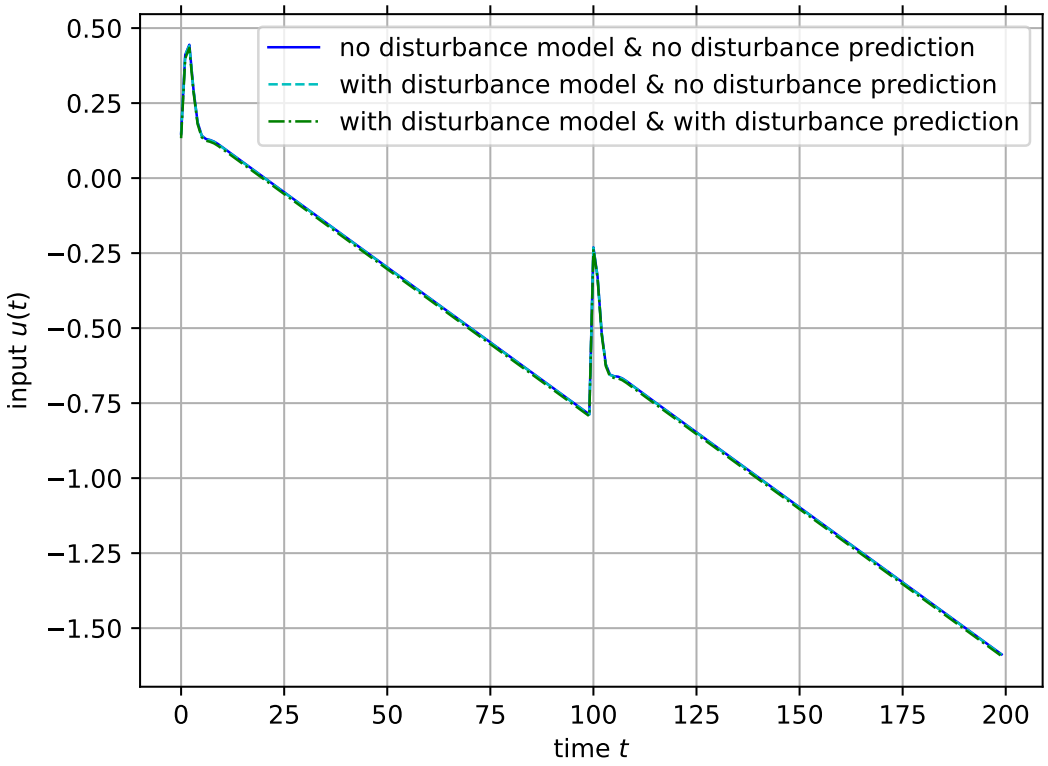


Figure 3.4: Numerical example: Input of a plant with GPC.

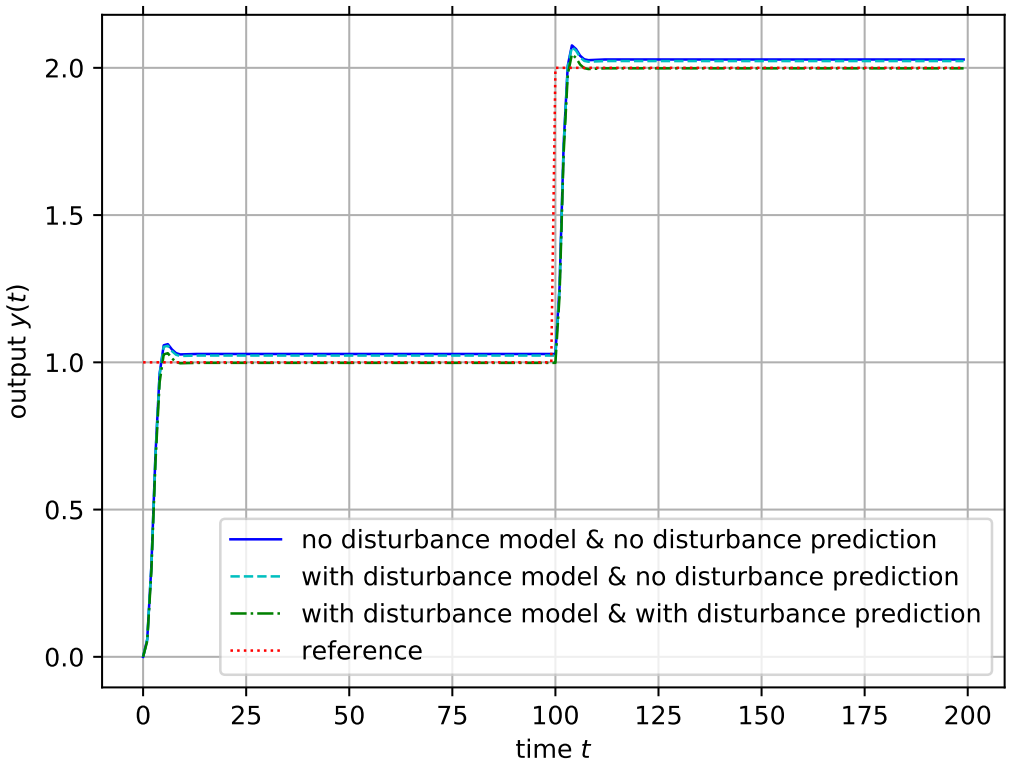


Figure 3.5: Numerical example: Output of a plant with GPC.

3.4 Simulation results

In this section, the dynamic model of a carbon capture plant is first identified and validated using neural networks. For the model identification issue, the inputs are the lean MEA flow rate $u(t)$ [14, 20], the flue gas flow rate (kg/s), $v_1(t)$, and the mass fraction of CO_2 in the flue gas, $v_2(t)$, while the output is the capture level (unitless), $y(t)$. Afterwards, some possible model-based control protocols focusing on the tracking problem of the carbon capture level $y(t)$ are applied on the identified neural-network-based PCC plant model. For this control problem, lean MEA flow rate is the only manipulated input where the other inputs (i.e., v_1(t) and v_2(t)) in model identification are some disturbance channels. In light of the GPC protocol, the controller performances are demonstrated and evaluated.

3.4.1 Model identification, order selection and validation

The data used for the plant model identification are generated by the first-principle PCC model [43] with the sampling time $T_s = 2.5$ s. During preprocessing, dc-offsets of both the input features $\mathbf{x}(t)$ and output $y(t)$ are removed. The model structures are neural networks parametrised by model parameter vector ϑ $\bar{D}$ whose dimension reflects unknown hidden layer size $\tilde{h}_2$ and model orders n_a, n_b, n_{d_1}, & n_{d_2} (Subsection 3.3.1). To reduce the number of candidate models, $n_b = n_{d_1} = n_{d_2}$ with the hidden layer size $\tilde{h}_2 = 1$ is assumed for the initial model order selection. Only n_a and n_b should be determined to fix $\bar{D}$. The training set contains 110 samples while the testing set contains 50 samples. With the scikit-learn tool box [82], the model is identified under the neural network framework. For both n_a and n_b ranging from 1 to 10, the model performances are quantised by AIC. Theoretically, the selected model orders should have the minimum AIC value (Figure 3.6), i.e., $n_a = 10$ and $n_b = 5$. The model order pair selected by AIC_c or Bayesian information criterion (BIC) [74] is $n_a = 5$ and $n_b = 5$.

Correspondingly, the selected candidate models must pass the whiteness and independence tests so as to validate their performances on approximating the first principle

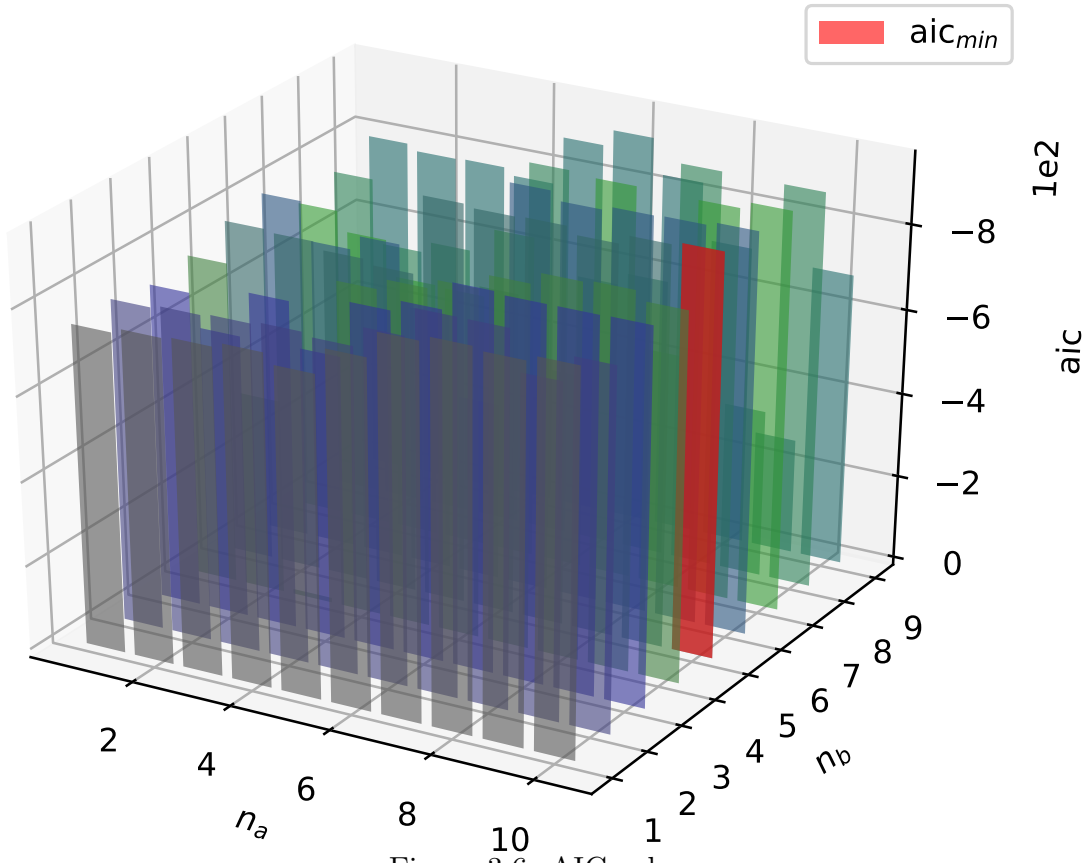


Figure 3.6: AIC values.

Table 3.1: Validated model orders and fit percentages.

(n_a, n_b)	$\tilde{h}_{2\min}$	fit (%)
(5, 5)	/	/
(7, 5)	3	97.77
(10, 1)	1	98.41
(10, 5)	1	98.42

Table 3.2: Selected model order pairs.

Model selection criteria	n_a	n_b
AIC	10	5
AIC$_c$	5	5
BIC	5	5

PCC model [43]. Validation is then applied based on the residual analysis in Subsection 3.2.5. The tests are conducted not only for the models selected by AIC, AIC$_c$ or BIC, but the candidate models with orders around the neighbours of the criterion-based ones, i.e., n_a and n_b are searched within {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. The hidden layer size $\tilde{h}_2$ is enumerated from 1 to 10. For each specified $\tilde{h}_2$ and n_a-n_b pair, a

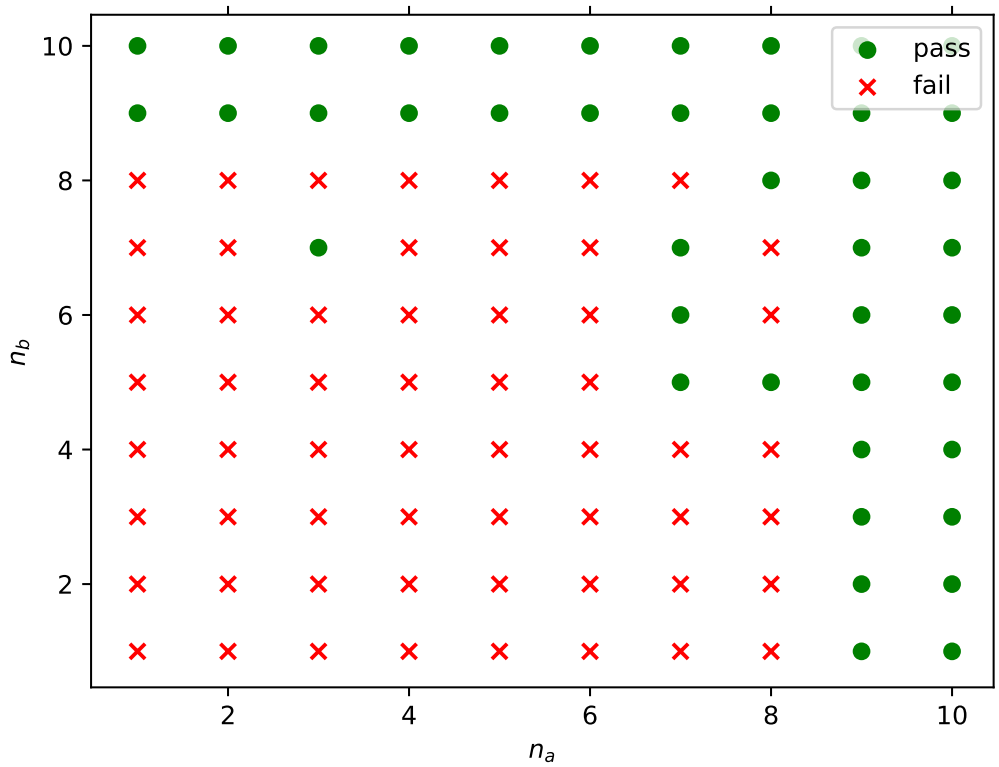


Figure 3.7: A set of whiteness and independence tests.

validated model must meet two constraints: (a) It can achieve a good fit (over 90% fit) with the observed data generated by the first-principle model; (b) the residual $\epsilon(t)$ of the candidate model can pass whiteness and independence tests. If the whiteness and independent tests can be passed for the residuals $\epsilon(t)$ of a candidate model (i.e., a predictor model (Eq. (3.14) with specified orders n_a & n_b, hidden layer size $\tilde{h}_2$, and the identified parameters θ), the white Gaussian noise assumption of the prediction error (Eq. (3.17)) for that candidate model is valid. In this simulation, for a specific n_a-n_b pair, if there exists any $\tilde{h}_2 \in \{1, 2, \cdots, 10\}$ such that the identified predictor model can pass the whiteness and independence tests, this n_a-n_b pair is recorded with "pass" (Figure 3.7). Although the model order pair, $n_a = 5$ and $n_b = 5$, is selected by AIC_c or BIC, the corresponding candidate model fails the tests (Figure 3.7). Table 3.1 only gives the smallest hidden layer sizes $\tilde{h}_{2\text{min}}$ with respect to some typical model order pairs (determined by AIC, AIC_c, BIC, etc.) such that the candidate models can pass the whiteness and independence tests. It is observed that if the model has passed the tests, the fit percentage is generally over 90%. Instead of the above constraints for

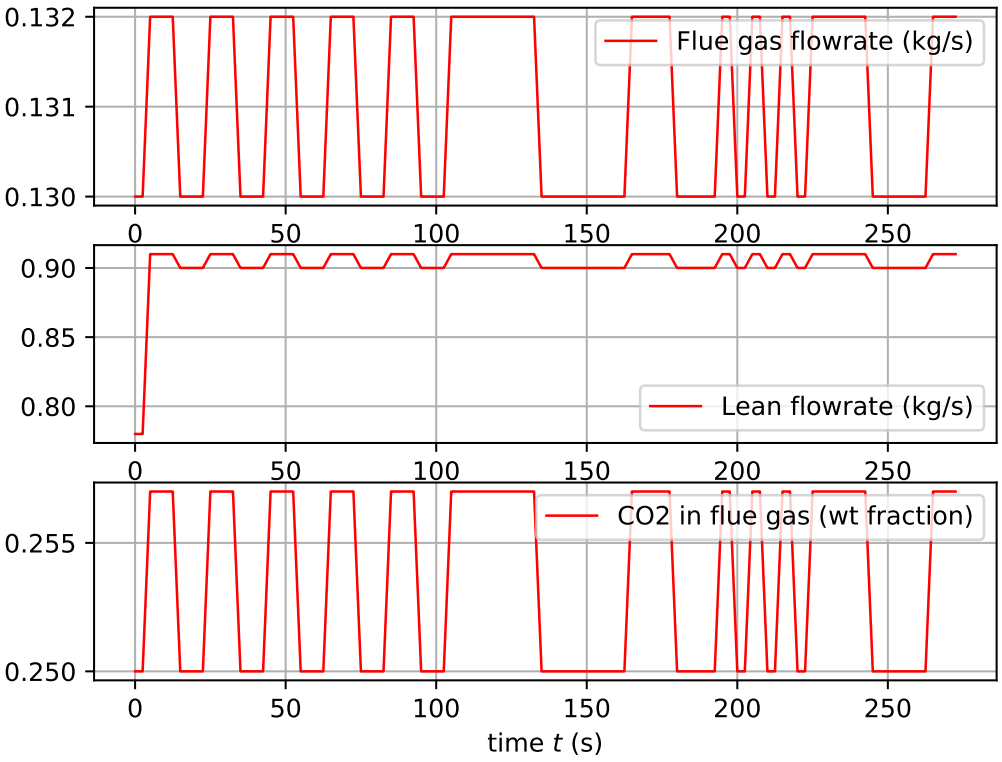


Figure 3.8: Historical inputs for model identification.

validated models, the number of model parameters $\bar{D}$ is further considered to avoid over-fitting. A candidate model with $n_a = 10$, $n_b = 1$, and $\tilde{h}_2 = 1$ is finally selected since $\bar{D} = (\tilde{h}_1 + 2) \cdot \tilde{h}_2 + 2 = 17$ is the smallest among all the validated models. Note that all the predictor models in the model set (Eq. (3.14)) can only achieve one-step-ahead prediciton. The reason is that the control protocols used afterwards are based on one-step-ahead algorithms (Eqs. (3.91) and (4.33)). According to input and output dynamics (Figure 3.8 and Figure 3.9), the fit percentage of the selected model is 98.41% for the one-step-ahead prediction. Furthermore, the selected model also has reasonable performance on the multi-step-ahead prediction. The fit percentage for the carbon capture level is 93.43%. This value is lower than 98.41% of the one-step-ahead prediction but still well above 90%. The residual analysis (Figures 3.10 and 3.11) of the model indicates $\zeta_1(\tau)$ and $\zeta_2(\tau)$ are within the 99% confidence intervals.

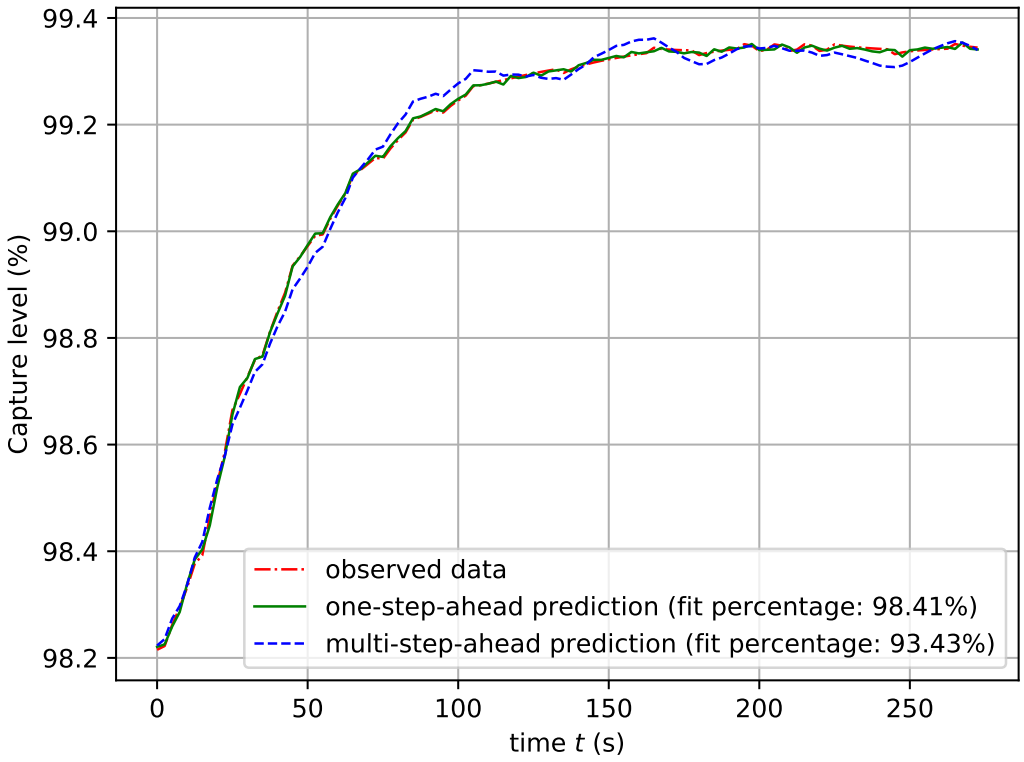


Figure 3.9: Output dynamics.

3.4.2 Model-based control

Note that the controlled nonlinear PCC plant is the validated neural network model selected in Subsection 3.4.1. The prediction model (Eq. (3.40)) is linearised based on this nonlinear model using the first-order Taylor approximation so as to derive $A(q^{-1})$, $B(q^{-1})$ and $D(q^{-1})$. This linearisation includes two parts: First, the equilibrium point of the neural-network model is iteratively derived. Secondly, a linear model is obtained around this equilibrium point. We consider a general MIMO case. The equilibrium point of the model is derived as follows. Define

$$\mathbf{y}(t) \triangleq (y_1(t), y_2(t), \cdots, y_{\bar{n}}(t))^T \in \mathbb{R}^{\bar{n}} \tag{3.93}$$

$$\mathbf{u}(t) \triangleq (u_1(t), u_2(t), \cdots, u_{\bar{m}}(t))^T \in \mathbb{R}^{\bar{m}}. \tag{3.94}$$

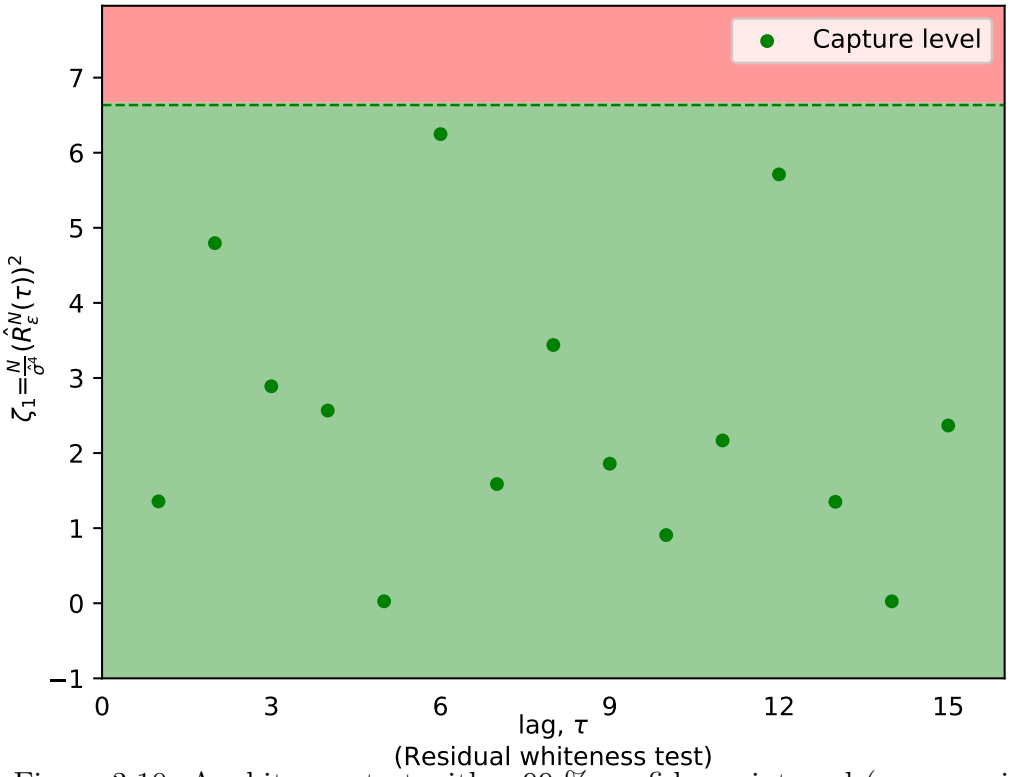


Figure 3.10: A whiteness test with a 99 % confidence interval (green region).

The input feature can be then represented by

$$\begin{aligned}
\mathbf{x}(t) \triangleq &(y_1(t-1), \cdots, y_1(t-n_a[1,1]), \cdots, y_{\bar{n}}(t-1), \cdots, y_{\bar{n}}(t-n_a[1,\bar{n}]),\\
&u_1(t-1), \cdots, u_1(t-n_b[1,1]), \cdots, u_{\bar{m}}(t-1), \cdots, u_{\bar{m}}(t-n_b[1,\bar{m}]),\\
&y_1(t-1), \cdots, y_1(t-n_a[2,1]), \cdots, y_{\bar{n}}(t-1), \cdots, y_{\bar{n}}(t-n_a[2,\bar{n}]),\\
&u_1(t-1), \cdots, u_1(t-n_b[2,1]), \cdots, u_{\bar{m}}(t-1), \cdots, u_{\bar{m}}(t-n_b[2,\bar{m}]),\\
&\cdots,\\
&y_1(t-1), \cdots, y_1(t-n_a[\bar{n},1]), \cdots, y_{\bar{n}}(t-1), \cdots, y_{\bar{n}}(t-n_a[\bar{n},\bar{n}]),\\
&u_1(t-1), \cdots, u_1(t-n_b[\bar{n},1]), \cdots, u_{\bar{m}}(t-1), \cdots, u_{\bar{m}}(t-n_b[\bar{n},\bar{m}]))^T\\
\triangleq &(\mathbf{x}_1^T(t), \mathbf{x}_2^T(t), \cdots, \mathbf{x}_{\bar{n}}^T(t))^T
\end{aligned} \tag{3.95}$$

$$\begin{aligned}
\mathbf{x}_k(t) \triangleq &(y_1(t-1), \cdots, y_1(t-n_a[k,1]), \cdots, y_{\bar{n}}(t-1), \cdots, y_{\bar{n}}(t-n_a[k,\bar{n}]),\\
&u_1(t-1), \cdots, u_1(t-n_b[k,1]), \cdots, u_{\bar{m}}(t-1), \cdots, u_{\bar{m}}(t-n_b[k,\bar{m}]))^T\\
\triangleq &(\bar{\mathbf{y}}_k^T(t), \bar{\mathbf{u}}_k^T(t))^T
\end{aligned} \tag{3.96}$$

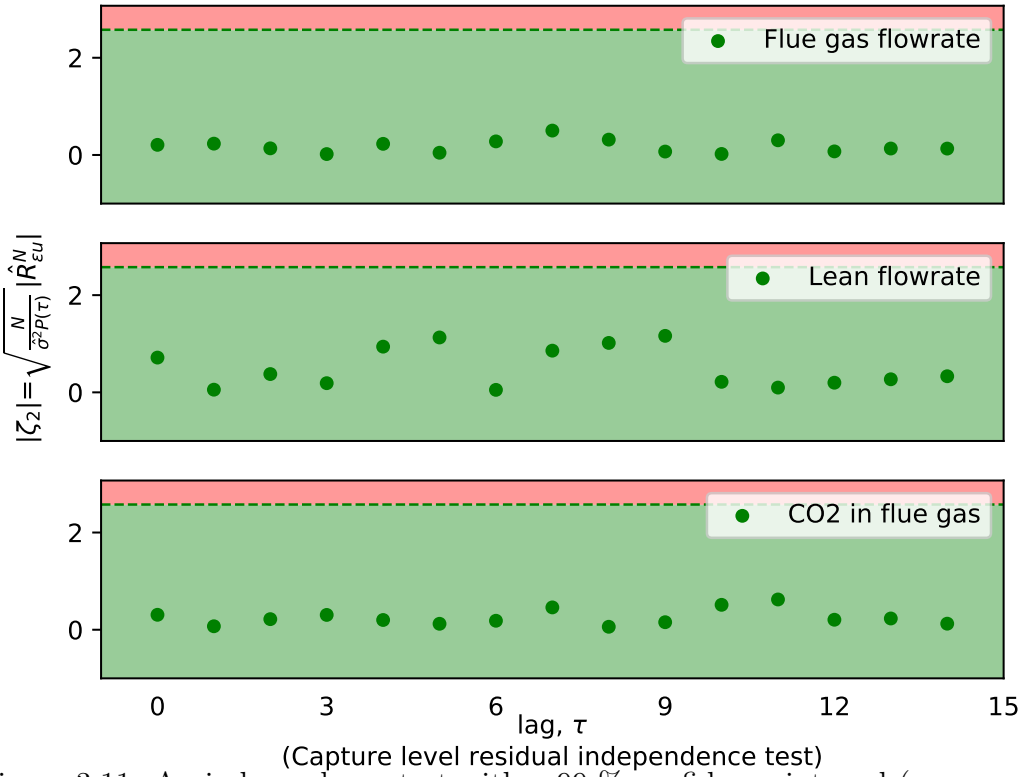


Figure 3.11: An independence test with a 99 % confidence interval (green region).

with n_a and n_b being $\bar{n} \times \bar{n}$ and $\bar{n} \times \bar{m}$ integer matrix, respectively. Compared with Eq. (3.7), the predictor model w.r.t. each output can be written as

$$\hat{y}_k(t|t-1) = g_k(\mathbf{y}^{t-1}, \mathbf{u}^{t-1}; \theta_k) = \tilde{g}_k(\mathbf{x}_k(t); \theta_k) = \tilde{g}_k(\bar{\mathbf{y}}_k(t), \bar{\mathbf{u}}_k(t); \theta_k)$$

i.e.,

$$\mathbf{y}(t) = \tilde{g}(\mathbf{x}(t); \theta) = \tilde{g}(\bar{\mathbf{y}}(t), \bar{\mathbf{u}}(t); \theta) \tag{3.97}$$

for the MIMO representation. In the steady state, $y_k(t) = y_{\mathrm{ss}k}$ and $u_j(t) = u_{\mathrm{ss}j}$ for $\forall t, j \in \{1, \cdots, \bar{n}\}, k \in \{1, \cdots, \bar{m}\}$ such that the above equation can be reformatted as

$$\mathbf{y}_{\mathrm{ss}} = \tilde{g}_{\mathrm{ss}}(\mathbf{x}_{\mathrm{ss}}; \theta_{\mathrm{ss}}) = \tilde{g}_{\mathrm{ss}}(\mathbf{y}_{\mathrm{ss}}, \mathbf{u}_{\mathrm{ss}}; \theta_{\mathrm{ss}}) \tag{3.98}$$

where

$$\mathbf{u}_{\mathrm{ss}} \triangleq (u_{\mathrm{ss}1}, \cdots, u_{\mathrm{ss}\bar{m}})^T \tag{3.99}$$

$$\mathbf{y}_{\mathrm{ss}} \triangleq (y_{\mathrm{ss}1}, \cdots, y_{\mathrm{ss}\bar{n}})^T \tag{3.100}$$

$$\mathbf{x}_{\mathrm{ss}} \triangleq (\mathbf{y}_{\mathrm{ss}}^T, \mathbf{u}_{\mathrm{ss}}^T)^T. \tag{3.101}$$

Given some specified steady-state input $\mathbf{u}_{\rm ss}$, the output $\mathbf{y}_{\rm ss}$ in Eq. (3.98) can be iteratively estimated with Newton's method, which means

$$\begin{aligned}&\tilde{g}_{\rm ss}(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss}) - \mathbf{y}_{\rm ss}^n + (\tilde{g}_{\rm ss}'^T(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss}) - I)(\mathbf{y}_{\rm ss}^{n+1} - \mathbf{y}_{\rm ss}^n) = 0 \\ &\Rightarrow \mathbf{y}_{\rm ss}^{n+1} = (\tilde{g}_{\rm ss}'^T(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss}) - I)^{-1}(\tilde{g}_{\rm ss}'^T(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss})\mathbf{y}_{\rm ss}^n - \tilde{g}_{\rm ss}(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss}))\end{aligned} \tag{3.102}$$

where $\mathbf{y}_{\rm ss}^n$ is the nth estimate of $\mathbf{y}_{\rm ss}$ and

$$\tilde{g}_{\rm ss}'(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss}) \triangleq \frac{\partial \tilde{g}_{\rm ss}(\mathbf{y}_{\rm ss}, \mathbf{u}_{\rm ss}; \theta_{\rm ss})}{\partial \mathbf{y}_{\rm ss}}|_{\mathbf{y}_{\rm ss}=\mathbf{y}_{\rm ss}^n}. \tag{3.103}$$

where $\tilde{g}_{\rm ss}'(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss})$ is computed as follows. On the basis of the definition in Eq. (3.96), further suppose

$$\begin{aligned}\mathbf{x}_k(t) = (&y_{\rm ss1}, \cdots, y_{\rm ss1}, \cdots, y_{\rm ss\bar{n}}, \cdots, y_{\rm ss\bar{n}}, \\ &u_{\rm ss1}, \cdots, u_{\rm ss1}, \cdots, u_{\rm ss\bar{m}}, \cdots, u_{\rm ss\bar{m}})^T \triangleq (\bar{\mathbf{y}}_{{\rm ss}k}, \bar{\mathbf{u}}_{{\rm ss}k})^T\end{aligned} \tag{3.104}$$

where $\bar{\mathbf{y}}_{{\rm ss}k} \in \mathbb{R}^{\sum_{j=1}^{\bar{n}} n_a[k,j]}$ and $\bar{\mathbf{u}}_{{\rm ss}k} \in \mathbb{R}^{\sum_{j=1}^{\bar{m}} n_b[k,j]}$. Eqs. (3.97), (3.98) and (3.104) give $\tilde{g}_{{\rm ss}k}(\mathbf{y}_{\rm ss}, \mathbf{u}_{\rm ss}; \theta_{\rm ss}) = \tilde{g}_k(\bar{\mathbf{y}}_k(t), \bar{\mathbf{u}}_{{\rm ss}k}; \theta_k)$, which implies

$$\frac{\partial \tilde{g}_{{\rm ss}k}(\mathbf{y}_{\rm ss}, \mathbf{u}_{\rm ss}; \theta_{\rm ss})}{\partial \mathbf{y}_{\rm ss}} = \frac{\partial \bar{\mathbf{y}}_k(t)}{\partial \mathbf{y}_{\rm ss}} \frac{\partial \tilde{g}_k(\bar{\mathbf{y}}_k(t), \bar{\mathbf{u}}_{{\rm ss}k}; \theta_k)}{\partial \bar{\mathbf{y}}_k(t)} \tag{3.105}$$

where

$$\frac{\partial \tilde{g}_k(\bar{\mathbf{y}}_k(t), \bar{\mathbf{u}}_{{\rm ss}k}; \theta_k)}{\partial \bar{\mathbf{y}}_k(t)} = \frac{\partial \mathbf{a}^2(t)}{\partial \bar{\mathbf{y}}_k(t)} \cdots \frac{\partial \mathbf{a}^{\bar{l}-2}(t)}{\partial \mathbf{a}^{\bar{l}-3}(t)} \frac{\partial \mathbf{a}^{\bar{l}-1}(t)}{\partial \mathbf{a}^{\bar{l}-2}(t)} \frac{\partial \tilde{g}_k(\bar{\mathbf{y}}_k(t), \bar{\mathbf{u}}_{{\rm ss}k}; \theta_k)}{\partial \mathbf{a}^{\bar{l}-1}(t)} \tag{3.106}$$

and

$$\frac{\partial \mathbf{a}^l(t)}{\partial \mathbf{a}^{l-1}(t)} = \frac{\partial \sigma_{\rm a}(\mathbf{W}^l \mathbf{a}^{l-1}(t) + \mathbf{b}^l)}{\partial \mathbf{a}^{l-1}(t)} = (\mathbf{W}^l)^T \sigma_{\rm a}'(\mathbf{W}^l \mathbf{a}^{l-1}(t) + \mathbf{b}^l). \tag{3.107}$$

Given $\mathbf{u}_{\rm ss}$, with $\mathbf{y}_{\rm ss}^n$ as the nth estimate of the equilibrium point $\mathbf{y}_{\rm ss}$, we can construct the associated $\bar{\mathbf{y}}_{{\rm ss}k}^n$ as the nth estimate of $\bar{\mathbf{y}}_{{\rm ss}k}$ for $\forall k$. Thus, with $\mathbf{y}_{\rm ss} = \mathbf{y}_{\rm ss}^n$ and $\bar{\mathbf{y}}_k(t) = \bar{\mathbf{y}}_{{\rm ss}k}^n$, Eqs. (3.105), (3.106) and (3.107) give the value of $\frac{\partial \tilde{g}_{\rm ss}(\mathbf{y}_{\rm ss}, \mathbf{u}_{\rm ss}; \theta_{\rm ss})}{\partial \mathbf{y}_{\rm ss}}|_{\mathbf{y}_{\rm ss}=\mathbf{y}_{\rm ss}^n}$, i.e., $\tilde{g}_{\rm ss}'(\mathbf{y}_{\rm ss}^n, \mathbf{u}_{\rm ss}; \theta_{\rm ss})$ in Eq. (3.103). A linear model is then derived from Eq. (3.97)

$$\mathbf{y}(t) = \tilde{g}(\bar{\mathbf{y}}_{\rm ss}^n, \bar{\mathbf{u}}_{\rm ss}; \theta) + (\frac{\partial \tilde{g}(\bar{\mathbf{y}}(t), \bar{\mathbf{u}}(t); \theta)}{\partial (\bar{\mathbf{y}}^T(t), \bar{\mathbf{u}}^T(t))^T})^T \left(\begin{pmatrix} \bar{\mathbf{y}}(t) \\ \bar{\mathbf{u}}(t) \end{pmatrix} - \begin{pmatrix} \bar{\mathbf{y}}_{\rm ss}^n(t) \\ \bar{\mathbf{u}}_{\rm ss}(t) \end{pmatrix} \right) \tag{3.108}$$

which can be then used to derive the prediction model (Eq. (3.40)). These polynomials may generate model uncertainties due to plant nonlinearities. For the same input

dynamics (Fig. 3.8), there exist mismatches between the output responses of the prediction model, the neural-network-based plant model and the observed data samples from the first-principle model (Fig. 3.12). Based on the aforementioned prediciton model, tuning parameters such as the time horizon (N_r), and the weight matrices ($\mathbf{Q}$ and $\mathbf{R}$) should be specified by a user to achieve the specified control objective (Eq. (3.43)). N_r is the concerned time horizon. $\mathbf{Q}$ is the penalty of the tracking error (i.e., $r(t+k) - y(t+k)$) within the time horizon N_r. $\mathbf{R}$ is the penalty of the manipulated input deviation (i.e., $\Delta u(t+k) = u(t+k) - u(t+k-1)$) within the time horizon N_r. The control objective (Eq. (3.43)) indicates there should be trade-off between the tracking error and the input manipulation. For the smooth input dynamics, entries of $\mathbf{Q}$ should be large while those of $\mathbf{R}$ should be small. In contrast, for the fast output responses, entries of $\mathbf{Q}$ should be small while those of $\mathbf{R}$ should be large. The reference signal of the carbon capture level is generated identically to Figure 3.15.

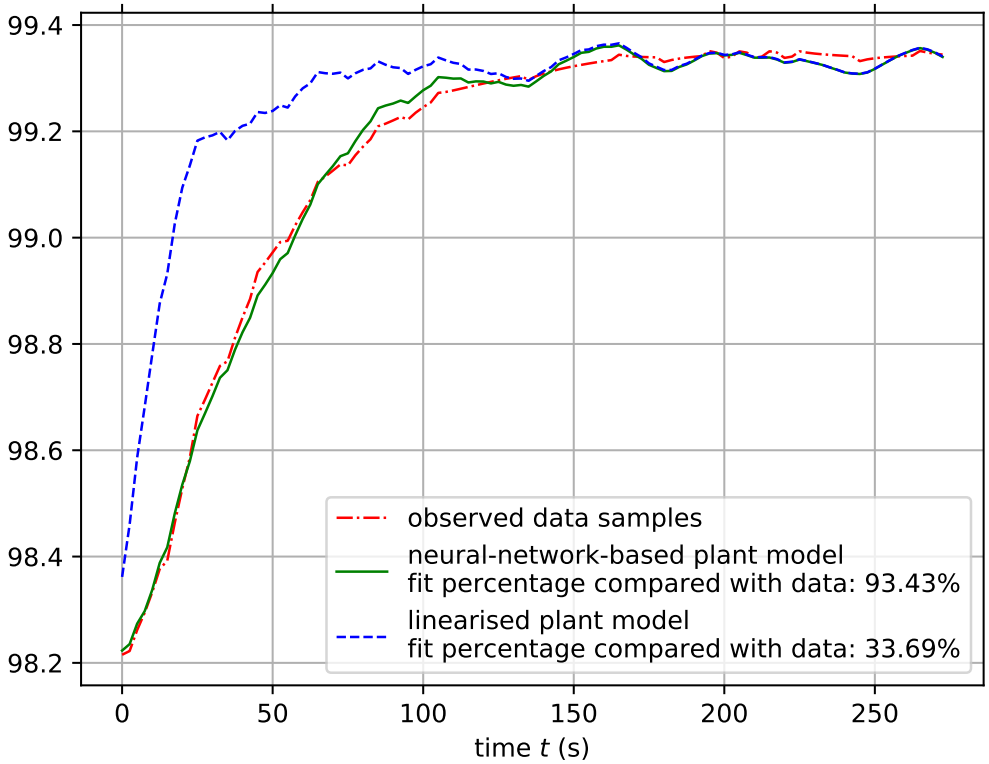


Figure 3.12: Comparison between observed data and model predictions.

In this case study, when no disturbances are applied, some good performance of GPC is obtained with the tuning parameters of $N_r = 3$, $\mathbf{Q} = 1 \cdot I_{N_r \times N_r}$ and

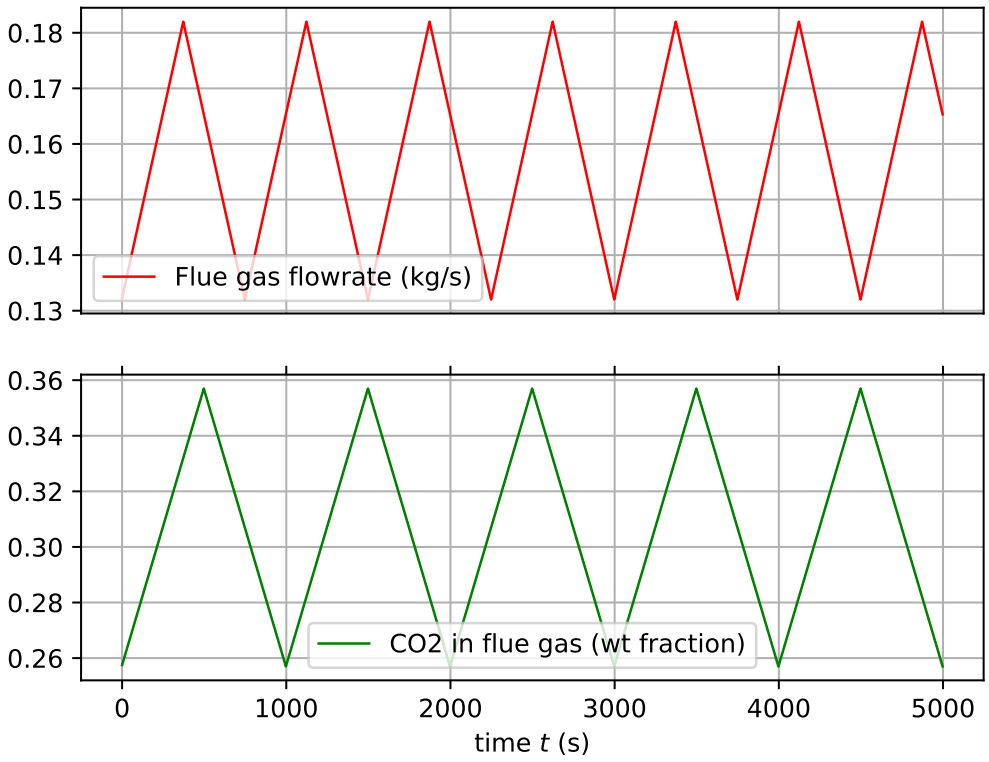


Figure 3.13: Disturbances.

$\mathbf{R} = 30 \cdot I_{N_r \times N_r}$ where $I_{N_r \times N_r} \in \mathbb{R}^{N_r \times N_r}$ is an identity matrix. Afterwards, the time-variant disturbances, i.e., the flue gas flow rate and the CO_2 mass fraction of the flue gas (Fig. 3.13), are applied to the controlled nonlinear PCC plant, which can be periodical ramp changes due to the variations of power generation [14]. Simultaneously, Fig. 3.15 shows no matter the disturbances (Figure 3.13) are applied or not, GPC can achieve some good performances for this tracking problem of carbon capture levels. However, due to the ramping disturbances applied, there exist some constant deviations which may not be eliminated. Nevertheless, an underline model should be identified before the tuning parameters of GPC can be tested online. The model not only lacks nonlinearities of the controlled plant but is usually obtained with offline sensitivity or identification tests as shown in Subsection 3.4.1, which makes the tuning procedure complicated.

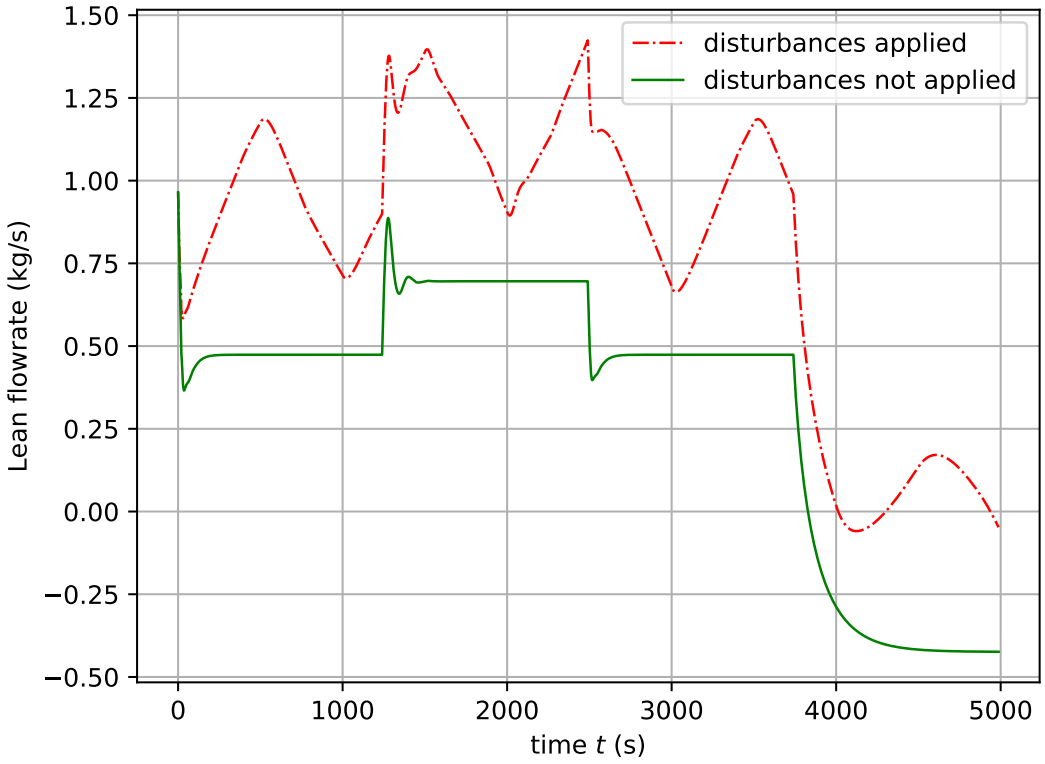


Figure 3.14: Manipulated inputs of GPC.

3.5 Summary

In this chapter the control issue of the MEA-based PCC plant is solved with model-based control, namely, GPC.

- We have identified a validated nonlinear PCC plant model using the data generated by a first-principle model. The model is identified with the back-propagation algorithm, approximately located by model order selection criteria such as AIC, AIC_c and BIC, and then searched around the neighbours of the criterion-determined model orders. The carbon capture plant described by the neural-network-based model can pass residual analysis and fit well with the data set.

- Subsequently, one advanced model-based controller termed as GPC has been applied to a identified neural-network-based carbon capture plant described by a neural-network-based model. GPC is a prevailing method which can be applied with only measured inputs and outputs as presented. Thus, no extra observers

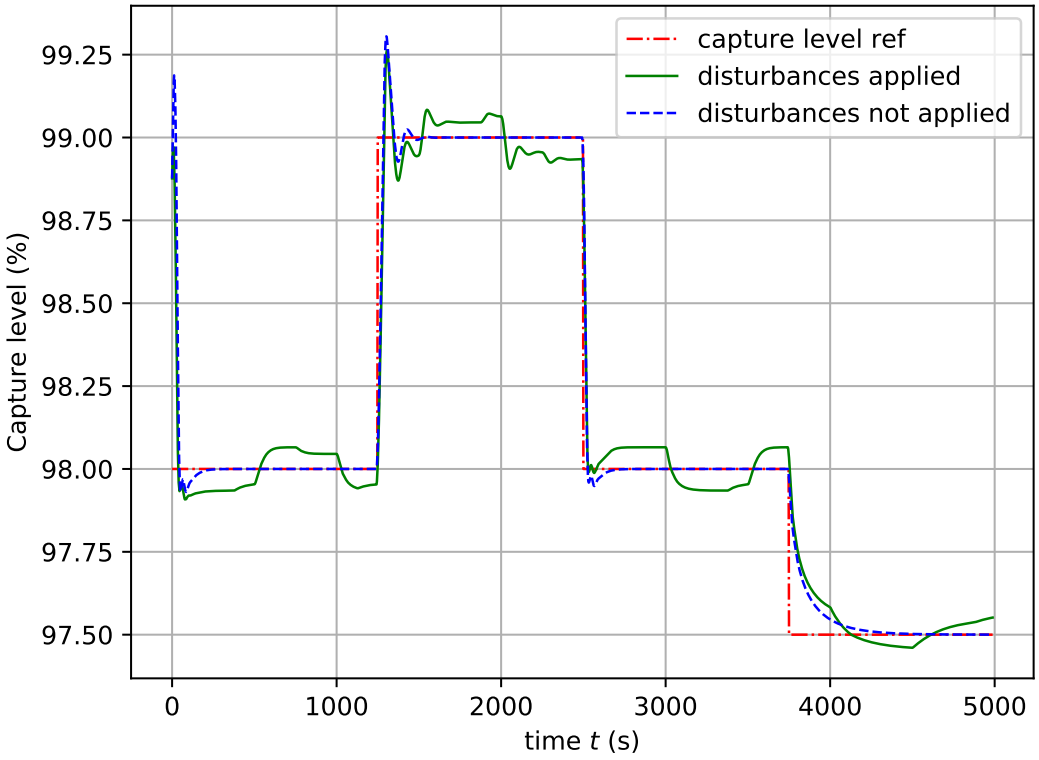


Figure 3.15: Controlled outputs of GPC.

are required for states of the carbon capture plant. However, when the disturbance dynamics of future are not compensated, due to those ramp changes of disturbances, there still exists some offset, i.e. a constant deviation of the output from the reference signal (Figure 3.15).

- For simplicity, the prediciton model of the MEA-based PCC plant in this chapter is a linear model derived from the neural-network-based plant model (Subsection 3.4.2). For a real system, it means the prediction model should be identified with the data from a real carbon capture plant. When the performances of GPC is unsatisfactory, some extra identification tests may be required to obtain a better prediction model, which is non-trivial.

Chapter 4

Model-free control of MEA-based post-combustion carbon capture process

4.1 Introduction

For the flexible operation of MEA-based PCC processes, recent studies concentrate on model-based protocols which require underline model parameters of carbon capture processes for controller design. The main focus is around the optimal operation utilising some model-based controls such as MPC. Nonetheless, for the design of a well-performed model-based controller, identification or sensitivity tests are needed before implementing such model-based controllers. This implies model identification, order selection and validation processes are necessary, which make it non-trivial for controller design. When the carbon capture plant is coupled with a power plant [14], uncertain conditions of the power plant may degrade dynamic performances of the carbon capture facilities. For instance, fluctuations of either the flue gas flow rate or the CO_2 mass fraction in the flue gas, dependent on the power plant load conditions, will change the operating point of the PCC process. These disturbances cause extra mismatches between the model and the controlled nonlinear PCC plant, which is classified

as model uncertainties. A large number of sensitivity or identification tests [20, 60] for different operating points of the controlled plant must be conducted before the controller can be properly tuned and implemented online. It makes the model-based controller design a non-trivial issue. In this chapter, a novel application of the MFAC algorithm is proposed that only uses measured input-output data for carbon capture processes. Compared with PI control, the stability of the closed-loop system can be easily guaranteed by increasing a stabilising parameter. By updating the pseudo-partial derivative vector to estimate a dynamic model of the controlled plant online, this new protocol is robust to plant uncertainties. Compared with MPC, tuning tests of the protocol can be conducted online without non-trivial repetitive offline sensitivity or identification tests. Performances of the MFAC protocol are demonstrated within a neural network carbon capture plant model, identified and validated with data generated by a first-principle carbon capture model.

This chapter is organised as follows. Firstly, the conventional model-free control protocol, PI and some preliminaries on MFAC are introduced. Secondly, a control problem similar to the one defined in Chapter 3 is formulated. Secondly, compared with the GPC and PI control protocols, MFAC is designed based on an iterative algorithm including online linear model update, control protocol implementation and a reset rule for model parameters. Thirdly, compared with PI control and GPC, performances of MFAC applied to the neural-network-based carbon capture plant model are demonstrated. A summary is given in the end.

4.2 Problem formulation

Similar to Section 3.3, we consider the tracking problem of carbon capture level $y(\mathrm{t})$ for the MEA-based carbon capture plant. The lean MEA flow rate $u(\mathrm{t})$ is the manipulated input and there exist two disturbance channels, $v_1(\mathrm{t})$ and $v_2(\mathrm{t})$. These disturbances may change the operating points of the carbon-capture process and deteriorate the performances of the controllers. Note that only the robustness to model uncertainties is demonstrated and evaluated in this chapter. Some model-free controllers are introduced to solve such a problem.

4.2.1 Simple model-free controller - PI control

PID is a simple model-free controller. A continuous-time PID controller can be represented by

$$u(t) = K_p e_c(t) + K_i \int_0^t e_c(\tau)d\tau + K_d \frac{de_c(t)}{dt} \tag{4.1}$$

where $e_c(t)$ is the controller input. K_p, K_i, and K_d are the proportional, integral, and derivative gains of the PID control, respectively. The integral and derivative parts can be approximated with

$$u(t) \approx u(t-1) + \frac{e_c(t) + e_c(t-1)}{2} T_s \tag{4.2}$$

and

$$u(t) \approx \frac{e_c(t) - e_c(t-1)}{T_s} \tag{4.3}$$

respectively. These approximations, thus, indicate

$$\frac{u(t)}{e_c(t)} = K_p + K_i \frac{T_s}{2} \frac{1+q^{-1}}{1-q^{-1}} + K_d \frac{1-q^{-1}}{T_s} \tag{4.4}$$

which can be further expanded as

$$\begin{aligned} u(t) =& u(t-1) + K_p(e_c(t) - e_c(t-1)) + K_i \frac{T_s}{2}(e_c(t) + e_c(t-1)) \\ &+ K_d \frac{1}{T_s}(e_c(t) - 2e_c(t-1) + e_c(t-2)) \\ =& u(t-1) + (K_p + \frac{K_i T_s}{2} + \frac{K_d}{T_s}) e_c(t) + (-K_p + \frac{K_i T_s}{2} - \frac{2K_d}{T_s}) e_c(t-1) \\ &+ \frac{K_d}{T_s} e_c(t-2). \end{aligned} \tag{4.5}$$

Eq. (4.5) is termed as a discrete PID controller. For simplicity, we only consider the PI control, which means $K_d = 0$. It is obvious that the derivation of such a control protocol involves no extra model-based information of a plant. Therefore, PID is actually a model-free controller. However, it lacks methods for solving nonlinearities of a plant. Similar to model-based controllers, gain scheduling should be applied to the plant for different operating points to deal with nonlinearity issues.

4.2.2 Preliminaries on MFAC

Another set of model-free controllers is called MFAC [30, 31]. Suppose the true MEA-based PCC plant without disturbance channels for v_1 and v_2 can be expressed as

$$y(t+1) = f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t), u(t-1), \cdots, u(t-n_u)). \tag{4.6}$$

By denoting the stacked input vector of the past as

$$\tilde{\mathbf{U}}(t) \triangleq (u(t), u(t-1), \cdots, u(t-L+1)) \tag{4.7}$$

where L is the initial time index of the manipulated input u(t) for MFAC, the control objective can be defined as

$$J_u \triangleq \|\bar{r}(t+1) - y(t+1)\|^2 + \tilde{\lambda}\|\Delta\tilde{\mathbf{U}}(t)\|^2 \tag{4.8}$$

where $\tilde{\lambda}$ is a tuning parameter which should be determined by a user. For an optimal control protocol, the following objective should be minimised

$$u(t) = \arg\min_{u(t)} J_u \tag{4.9}$$

to track the reference signal $\bar{r}(t)$ of carbon capture level and $\|\cdot\|$ is the 2-norm of a specified vector. The necessary condition for Eq. (4.9) is $\partial J_u/\partial u(t) = 0$. According to [30], two assumptions are needed before the derivation of the MFAC protocol from Eq. (4.9). The stability of the protocol, however, requires another assumption (**Assumption** 4.3.1) which will be shown later.

Assumption 4.2.1 The partial derivative of f_0 w.r.t. the manipulated input $u(t)$, $u(t-1)$, $\cdots$, $u(t-L+1)$, are continuous.

Assumption 4.2.2 The plant represented by Eq.(4.6) is generalised Lipschitz, i.e.,

$$\|\Delta y(t+1)\| \leq \bar{b}\|\Delta\tilde{\mathbf{U}}(t)\| \tag{4.10}$$

where $\bar{b}$ is a positive constant. We show the plant description (Eq. (4.6)) can be linearised as follows.

$$\begin{aligned}\Delta y(t+1) =& f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t), u(t-1), \cdots, u(t-n_u)) \\ & -f_0(y(t-1), y(t-2), \cdots, y(t-n_y-1), u(t-1), u(t-2), \cdots, u(t-n_u-1)) \\ =& f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t), u(t-1), \cdots, u(t-n_u)) \\ & -f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t-1), u(t-1), \cdots, u(t-n_u)) \\ & +f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t-1), u(t-1), \cdots, u(t-n_u)) \\ & -f_0(y(t-1), y(t-2), \cdots, y(t-n_y-1), u(t-1), u(t-2), \cdots, u(t-n_u-1)).\end{aligned} \tag{4.11}$$

Together with **Assumption** 4.2.1, the first two terms to the right side of Eq. (4.11) can be rewritten as

$$\begin{aligned}& f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t), u(t-1), \cdots, u(t-n_u)) \\ & - f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t-1), u(t-1), \cdots, u(t-n_u)) \\ =& \frac{\partial f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t), u(t-1), \cdots, u(t-n_u))}{\partial u(t)}|_{u(t)=\underline{u}} \cdot (u(t) - u(t-1)) \\ \triangleq& \frac{\partial f_0^*}{\partial u(t)} \cdot (u(t) - u(t-1)) = \frac{\partial f_0^*}{\partial u(t)} \cdot \Delta u(t)\end{aligned} \tag{4.12}$$

where based on the differential mean theorem [30], there exists some point $\underline{u}$ in the interval $[u(t), u(t-1)]$ making this equality hold. In addition, define

$$\begin{aligned}& \Psi_1(t) \triangleq \Psi_1(y(t), y(t-1), \cdots, y(t-n_y-1), u(t-1), u(t-2), \cdots, u(t-n_u-1)) \\ \triangleq& f_0(y(t), y(t-1), \cdots, y(t-n_y), u(t-1), u(t-1), \cdots, u(t-n_u)) \\ & - f_0(y(t-1), y(t-2), \cdots, y(t-n_y-1), u(t-1), u(t-2), \cdots, u(t-n_u-1))\end{aligned} \tag{4.13}$$

which is a simplified representation of the last two terms to the right-side of Eq. (4.11). Combined with Eq. (4.12) and Eq. (4.13), the plant (Eq.(4.11)) can be represented by

$$\Delta y(t+1) = \frac{\partial f_0^*}{\partial u(t)} \Delta u(t) + \Psi_1(t). \tag{4.14}$$

On the ground of Eq. (4.13), we further define

$$\begin{aligned}& \Psi_k(t) \triangleq \Psi_k(y(t), y(t-1), \cdots, y(t-n_y-1), u(t-k), u(t-k-1), \cdots, u(t-n_u-1)) \\ \triangleq& \Psi_{k-1}(y(t), y(t-1), \cdots, y(t-n_y-1), u(t-k), u(t-k), \cdots, u(t-n_u-1))\end{aligned} \tag{4.15}$$

which has continuous partial derivative w.r.t. $u(t-k)$ according to **Assumption** 4.2.1. When $k = 2$, Eq. (4.14) and Eq. (4.15) give

$$\begin{aligned}
\Delta y(t+1) =& \frac{\partial f_0^*}{\partial u(t)}\Delta u(t) + \Psi_1(y(t), y(t-1), \cdots, y(t-n_y), \\
& u(t-1), u(t-2), \cdots, u(t-n_u-1)) \\
=& \frac{\partial f_0^*}{\partial u(t)}\Delta u(t) \\
& + \Psi_1(y(t), y(t-1), \cdots, y(t-n_y), u(t-1), u(t-2), \cdots, u(t-n_u-1)) \\
& - \Psi_1(y(t), y(t-1), \cdots, y(t-n_y), u(t-2), u(t-2), \cdots, u(t-n_u-1)) \\
& + \Psi_1(y(t), y(t-1), \cdots, y(t-n_y), u(t-2), u(t-2), \cdots, u(t-n_u-1)) \\
=& \frac{\partial f_0^*}{\partial u(t)}\Delta u(t) + \frac{\partial \Psi_1^*}{\partial u(t-1)}\Delta u(t-1) \\
& + \Psi_1(y(t), y(t-1), \cdots, y(t-n_y), u(t-2), u(t-2), \cdots, u(t-n_u-1)) \\
=& \frac{\partial f_0^*}{\partial u(t)}\Delta u(t) + \frac{\partial \Psi_1^*}{\partial u(t-1)}\Delta u(t-1) + \Psi_2(t)
\end{aligned} \tag{4.16}$$

which infers that for any L

$$\begin{aligned}
\Delta y(t+1) =& \frac{\partial f_0^*}{\partial u(t)}\Delta u(t) + \frac{\partial \Psi_1^*}{\partial u(t-1)}\Delta u(t-1) + \frac{\partial \Psi_2^*}{\partial u(t-2)}\Delta u(t-2) + \cdots \\
& + \frac{\partial \Psi_{L-1}^*}{\partial u(t-L+1)}\Delta u(t-L+1) + \Psi_L \\
=& \Psi_L(t) + \begin{pmatrix} \frac{\partial f_0^*}{\partial u(t)} & \frac{\partial \Psi_1^*}{\partial u(t-1)} & \frac{\partial \Psi_2^*}{\partial u(t-2)} & \cdots & \frac{\partial \Psi_{L-1}^*}{\partial u(t-L+1)} \end{pmatrix} \cdot \Delta\tilde{\mathbf{U}}(t).
\end{aligned} \tag{4.17}$$

Note that $\Psi_L(t)$ can be expressed as

$$\Psi_L(t) = \tilde{\mathbf{H}}_L(t)\Delta\tilde{\mathbf{U}}(t) \tag{4.18}$$

where $\tilde{\mathbf{H}}_L \in \mathbb{R}^{1\times L}$ which should exist infinite solutions if $\|\Delta\tilde{\mathbf{U}}(t)\| \neq 0$. Hence, by defining the PPD vector as

$$\begin{aligned}
\Phi(t) \triangleq& \tilde{\mathbf{H}}(t) + \begin{pmatrix} \frac{\partial f_0^*}{\partial u(t)} & \frac{\partial \Psi_1^*}{\partial u(t-1)} & \frac{\partial \Psi_2^*}{\partial u(t-2)} & \cdots & \frac{\partial \Psi_{L-1}^*}{\partial u(t-L+1)} \end{pmatrix} \\
=& (\phi_1(t), \phi_2(t), \cdots, \phi_L(t)) \in \mathbb{R}^{1\times L}
\end{aligned} \tag{4.19}$$

Eq. (4.17) can be rewritten as

$$\Delta y(t+1) = \Phi(t)\Delta\tilde{\mathbf{U}}(t) \tag{4.20}$$

where the nonlinear plant (Eq. (4.6)) behaviours can be estimated through the time-varying row vector $\Phi(t)$ as above. Note that in literature, the linearisation method

above is termed as compact form dynamic linearisation (CFDL)-MFAC when $L = 1$, and termed as partial form dynamic linearisation (PFDL)-MFAC when $L > 1$.

4.3 Model-free adaptive control

The PCC process is commonly modelled by first-principle strategies such as equilibrium-based or rate-based approaches [11], which infers that the process involves nonlinearities. Note that the time-variant flue gas flow rate, $v_1(t)$ and the mass fraction of CO_2 in flue gas, $v_2(t)$ may cause variations of the process operating point. Thus, nonlinearities will lead to mismatches between the controlled plant and the underline model of the model-based controllers, such as GPC. The model-free protocol [31] can form a dynamic linear model online for the controlled nonlinear plant with a PPD vector $\Phi(t)$. No offline model parameters are required when the controller is implemented in real time. As the process operating point varies, $\Phi(t)$ adapts to the changes.

4.3.1 Derivation of MFAC

In this subsection, the control protocol is derived w.r.t. the problem (Eq. (4.9)) described above while a projection algorithm is introduced to estimate the aforementioned PPD vector $\Phi(t)$ thereafter.

The control protocol is derived as follows. Combining Eq. (4.8) and Eq. (4.20)

gives

$$\begin{aligned}
\frac{\partial J_u}{\partial u(t)} =& \frac{\|\bar{r}(t+1) - y(t+1) + y(t) - y(t)\|^2 + \tilde{\lambda}\|\Delta\tilde{\mathbf{U}}(t)\|^2}{\partial u(t)} \\
=& \frac{\|\bar{r}(t+1) - \Delta y(t+1) - y(t)\|^2}{\partial u(t)} + \frac{\tilde{\lambda}\partial\sum_{k=1}^{L}|\Delta u(t-k+1)|^2}{\partial u(t)} \\
=& \frac{\partial\Phi(t)\Delta\tilde{\mathbf{U}}(t)}{\partial u(t)} \cdot \frac{\partial\|\bar{r}(t+1) - \Phi(t)\Delta\tilde{\mathbf{U}}(t) - y(t)\|^2}{\partial\Phi(t)\Delta\tilde{\mathbf{U}}(t)} + 2\tilde{\lambda}\Delta u(t) \\
=& -2\phi_1(t)(\bar{r}(t+1) - y(t) - \sum_{k=1}^{L}\phi_k(t)\Delta u(t-k+1)) + 2\tilde{\lambda}\Delta u(t) \\
=& -2\phi_1(t)(\bar{r}(t+1) - y(t) - \sum_{k=2}^{L}\phi_k(t)\Delta u(t-k+1)) \\
& + 2(\tilde{\lambda} + |\phi_1(t)|^2)\Delta u(t).
\end{aligned} \tag{4.21}$$

As mentioned before, a necessary condition for the control protocol derivation is $\partial J_u/\partial u(t) = 0$. Therefore,

$$\begin{aligned}
u(t) =& u(t-1) + \frac{\phi_1(t)(\bar{r}(t+1) - y(t))}{\tilde{\lambda} + |\phi_1(t)|^2} \\
& - \frac{\phi_1(t)\sum_{k=2}^{L}\phi_k(t)\Delta u(t-k+1)}{\tilde{\lambda} + |\phi_1(t)|^2}.
\end{aligned} \tag{4.22}$$

Since the true PPD vector is unknown for MFAC, the PPD vector $\Phi(t)$ above is replaced by an estimate, $\hat{\Phi}(t) \triangleq (\hat{\phi}_1(t), \hat{\phi}_2(t), \cdots, \hat{\phi}_L(t)) \in \mathbb{R}^{1\times L}$, which infers

$$\begin{aligned}
u(t) =& u(t-1) + \frac{\hat{\phi}_1(t)(\bar{r}(t+1) - y(t))}{\tilde{\lambda} + |\hat{\phi}_1(t)|^2} \\
& - \frac{\hat{\phi}_1(t)\sum_{k=2}^{L}\hat{\phi}_k(t)\Delta u(t-k+1)}{\tilde{\lambda} + |\hat{\phi}_1(t)|^2}.
\end{aligned} \tag{4.23}$$

For the real-time estimation of plant model parameters $\Phi(t)$, an optimisation problem is formulated as

$$J_\Phi = \|\hat{\Phi}(t) - \hat{\Phi}(t-1)\|^2 \tag{4.24}$$

subject to

$$\Delta y(t) = \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1) \tag{4.25}$$

which can be solved by the modified projection algorithm [83] and it is briefly introduced as follows. In light of Eq. (4.24) and Eq. (4.25), the objective function can be reformulated as

$$J_\Phi = |\Delta y(t) - \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)|^2 + \mu\|\hat{\Phi}(t) - \hat{\Phi}(t-1)\|^2 \tag{4.26}$$

where $\mu \in \mathbb{R}$. Taking the derivative of it w.r.t. $\Phi^T(t)$ gives

$$\begin{aligned}\frac{\partial J_\Phi}{\partial \hat{\Phi}^T(t)} =& \frac{\partial \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)}{\partial \hat{\Phi}^T(t)} \cdot \frac{\partial |y(t) - y(t-1) - \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)|^2}{\partial \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)} \\ &+ 2\mu(\hat{\Phi}^T(t) - \hat{\Phi}^T(t-1)) \\ =& -2\Delta\tilde{\mathbf{U}}(t-1) \cdot (y(t) - y(t-1) - \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)) \\ &+ 2\mu(\hat{\Phi}^T(t) - \hat{\Phi}^T(t-1)).\end{aligned} \tag{4.27}$$

Letting Eq.(4.27) equal to zero derives

$$\begin{aligned}0 =& -2\Delta\tilde{\mathbf{U}}(t-1)(\Delta y(t) - \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)) + 2\mu(\hat{\Phi}^T(t) - \hat{\Phi}^T(t-1)) \\ =& -2\Delta\tilde{\mathbf{U}}(t-1)(\Delta y(t) - \hat{\Phi}(t-1)\Delta\tilde{\mathbf{U}}(t-1) + \hat{\Phi}(t-1)\Delta\tilde{\mathbf{U}}(t-1) \\ &- \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t-1)) + 2\mu(\hat{\Phi}^T(t) - \hat{\Phi}^T(t-1)) \\ =& -2\Delta\tilde{\mathbf{U}}(t-1)(\Delta y(t) - \hat{\Phi}(t-1)\Delta\tilde{\mathbf{U}}(t-1)) \\ &+ 2(\mu + \Delta\tilde{\mathbf{U}}(t-1)\Delta\tilde{\mathbf{U}}^T(t-1))(\hat{\Phi}^T(t) - \hat{\Phi}^T(t-1))\end{aligned} \tag{4.28}$$

which can be further transformed into

$$\begin{aligned}\hat{\Phi}^T(t) =& \hat{\Phi}^T(t-1) + (\mu + \Delta\tilde{\mathbf{U}}(t-1)\Delta\tilde{\mathbf{U}}^T(t-1))^{-1}\Delta\tilde{\mathbf{U}}(t-1) \\ &\cdot (\Delta y(t) - \hat{\Phi}(t-1)\Delta\tilde{\mathbf{U}}(t-1)).\end{aligned} \tag{4.29}$$

With the matrix inversion lemma [78], Eq. (4.29) can be written as

$$\begin{aligned}\hat{\Phi}(t) =& \hat{\Phi}(t-1) \\ &+ \frac{(\Delta y(t) - \hat{\Phi}(t-1)\Delta\tilde{\mathbf{U}}(t-1))\Delta\tilde{\mathbf{U}}^T(t-1)}{\mu + \|\Delta\tilde{\mathbf{U}}(t-1)\|^2}.\end{aligned} \tag{4.30}$$

Currently, considering no stability issues, Eq. (4.30) and Eq. (4.23) give the recursive algorithm for the model update and control protocol implementation, respectively.

4.3.2 Implementation of MFAC

For both the flexibility and stability of the MFAC protocol, some additional tuning parameters η, $\boldsymbol{\rho}$ and $\bar{b}_1$ should be included for MFAC. The aforementioned model update (Eq. (4.30)) is modified as

$$\begin{aligned}\hat{\Phi}(t) =&\hat{\Phi}(t-1)\\ &+\frac{\eta(\Delta y(t)-\hat{\Phi}(t-1)\Delta\tilde{\mathbf{U}}(t-1))\Delta\tilde{\mathbf{U}}^T(t-1)}{\mu+\|\Delta\tilde{\mathbf{U}}(t-1)\|^2}\end{aligned} \tag{4.31}$$

a reset rule for the model parameter is introduced by

$$\hat{\phi}_1(t)=\hat{\phi}_1(1),\ \text{if}\ |\hat{\phi}_1(t)|<\bar{b}_1,\text{or}\ \text{sign}(\hat{\phi}_1(t))\neq\text{sign}(\hat{\phi}_1(1)) \tag{4.32}$$

and the control protocol implementation (Eq. (4.23)) is transformed into

$$\begin{aligned}u(t) =&u(t-1)+\frac{\rho_1\hat{\phi}_1(t)(\bar{r}(t+1)-y(t))}{\tilde{\lambda}+|\hat{\phi}_1(t)|^2}\\ &-\frac{\hat{\phi}_1(t)\sum_{k=2}^{L}\rho_k\hat{\phi}_k(t)\Delta u(t-k+1)}{\tilde{\lambda}+|\hat{\phi}_1(t)|^2}.\end{aligned} \tag{4.33}$$

Eq. (4.31), Eq. (4.32), and Eq. (4.33) form the iterative algorithm of the PFDL-MFAC protocol [30]. To apply this algorithm, tuning parameters within constraints (i.e., $\eta\in(0,1)$, $\mu>0$, $\boldsymbol{\rho}=(\rho_1,\,\rho_2,\,\cdots,\,\rho_L)^T$ with $\rho_k\in(0,1)$ for any k, $\tilde{\lambda}>\tilde{\lambda}_{\min}>0$, $\alpha>1$, and $b>0$) should be determined by the user. η and μ are related to the adaptive performances of the dynamic linear model for the controlled PCC plant. $\boldsymbol{\rho}$ and $\tilde{\lambda}$ are related to the control performances for the plant. For fast responses, η and $\boldsymbol{\rho}$ should increase while for smooth dynamics, μ and $\tilde{\lambda}$ should increase. The PPD vector $\hat{\Phi}(t)$ is updated online without using any prior knowledge of the offline model, which implies the iterative algorithm is model-free. Arbitrary initial conditions of $\hat{\Phi}(t=1)$ should be specified to set up the iteration.

When $L=1$, the above PFDL-MFAC protocol is reduced to the CFDL-MFAC,

which is presented by Eq. (4.34), Eq. (4.35) and Eq. (4.36).

$$\begin{aligned}\hat{\phi}_1(t) =& \hat{\phi}_1(t-1) \\ &+ \frac{\eta(\Delta y(t) - \hat{\phi}_1(t-1)\Delta u(t-1))\Delta u(t-1)}{\mu + |\Delta u(t-1)|^2}\end{aligned} \tag{4.34}$$

$$\hat{\phi}_1(t) = \hat{\phi}_1(1), \text{ if } |\hat{\phi}_1(t)| < \bar{b}_1, \text{or sign}(\hat{\phi}_1(t)) \neq \text{sign}(\hat{\phi}_1(1)) \tag{4.35}$$

$$\begin{aligned}u(t) =& u(t-1) + \frac{\rho_1 \hat{\phi}_1(t)(\bar{r}(t+1) - y(t))}{\tilde{\lambda} + |\hat{\phi}_1(t)|^2} \\ &- \frac{\hat{\phi}_1(t) \sum_{k=2}^{L} \rho_k \hat{\phi}_k(t) \Delta u(t-k+1)}{\tilde{\lambda} + |\hat{\phi}_1(t)|^2}.\end{aligned} \tag{4.36}$$

Compared with PI control, the above iterative method is easy to guarantee stability. If the closed-loop system is unstable or marginally stable, only the stabilising parameter $\tilde{\lambda}$ should increase for the stabilisation while PI control requires stability analysis such as the Nyquist criterion to determine whether to increase or decrease tuning parameters. In addition, the Nyquist criterion is a model-based method requiring model parameters. Furthermore, PI control is generally designed around fixed operating points while MFAC forms an adaptive dynamic linear model using online model update (Eq. (4.31)), i.e., MFAC already considers model uncertainties and should have strong robustness.

Compared with GPC requiring a prediction model, MFAC can be easily tuned online with measured input-output data of the controlled plant. If the underline model is inaccurate, the performances of GPC will be deteriorated. For the PCC process which is sensitive to ambient environments and is nonlinear, a large number of sensitivity or identification tests should be conducted around different operating points of the controlled plant before the controller can be applied online. MFAC only uses input-output data of the PCC plant. No offline model parameters are necessary for the online control implementation. The identified mathematical model of the PCC process is only used for the initial offline tuning. Afterwards, if the control performance is unsatisfactory, MFAC can be retuned online [30] without offline models. However, if the control performance of a model-based controller is poor, the model may be re-identified offline based on new data generated by the first-principle model, which is non-trivial. Therefore, the implementation of MFAC is easier.

4.3.3 Stability of MFAC

We give another assumption required for the stability of the MFAC protocol.

Assumption 4.3.1 The first element of PPD vector $\phi_1(t) > \xi$ ($\phi_1(t) > \xi$) for all k, where ξ is an arbitrary small positive constant.

The detailed proof of such an MFAC algorithm is shown by Hou et al. [31] based on **Assumption** 4.2.1, **Assumption** 4.2.2, and **Assumption** 4.3.1. Generally, the proof of stability includes two steps. First, the boundedness of the PPD vector should be proved. Second, the convergence of the tacking error and the bounded-input-bounded-output (BIBO) property of the closed-loop system should be demonstrated.

4.3.4 MFAC for MIMO system

Note that the aforementioned SISO MFAC protocol using PFDL (Eqs. (4.31), (4.32), and (4.33)) or CFDL (Eqs. (4.34), (4.35), and (4.36)), can be extended to an MIMO case [30], which implies the aforementioned linearisation (Eq. (4.25)), i.e.,

$$\Delta y(t+1) = \hat{\Phi}(t)\Delta\tilde{\mathbf{U}}(t)$$

should have

$$\tilde{\mathbf{U}}(t) = (u^T(t), u^T(t-1), \cdots, u^T(t-L+1))^T \in \mathbb{R}^{L\bar{m}} \tag{4.37}$$

$$\hat{\Phi}(t) = (\hat{\phi}_1(t), \hat{\phi}_2(t), \cdots, \hat{\phi}_L(t)) \in \mathbb{R}^{\bar{m}\times L\bar{m}} \tag{4.38}$$

$$\hat{\phi}_k(t) = \begin{pmatrix} \hat{\phi}^k_{1,1}(t) & \hat{\phi}^k_{1,2}(t) & \cdots & \hat{\phi}^k_{1,\bar{m}}(t) \\ \hat{\phi}^k_{2,1}(t) & \hat{\phi}^k_{2,2}(t) & \cdots & \hat{\phi}^k_{1,\bar{m}}(t) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\phi}^k_{\bar{m},1}(t) & \hat{\phi}^k_{\bar{m},2}(t) & \cdots & \hat{\phi}^k_{\bar{m},\bar{m}}(t) \end{pmatrix} \tag{4.39}$$

for $u, y \in \mathbb{R}^{\bar{m}}$ and $L \in \mathbb{Z}^+$. Thus, the CFDL- or PFDL-MFAC protocol for an MIMO system is presented as follows.

MIMO CFDL-MFAC

$$\begin{aligned}\hat{\Phi}(t) =&\hat{\Phi}(t-1)\\ &+\frac{\eta(\Delta y(t)-\hat{\Phi}^T(t-1)\Delta u(t-1))\Delta u^T(t-1)}{\mu+\|\Delta u(t-1)\|^2}\end{aligned} \tag{4.40}$$

$$\begin{aligned}\hat{\phi}^1_{i,i}(t) = \hat{\phi}^1_{i,i}(1),\ &\text{if}\ |\hat{\phi}^1_{i,i}(t)|<\bar{b}_1,\ \text{or}\ |\hat{\phi}^1_{i,i}(t)|>\alpha\bar{b}_1,\\ &\text{or}\ \text{sign}(\hat{\phi}^1_{i,i}(t))\neq\text{sign}(\hat{\phi}^1_{i,i}(1))\end{aligned} \tag{4.41}$$

$$\begin{aligned}\hat{\phi}^1_{i,j}(t) = \hat{\phi}^1_{i,j}(1),\ &\text{if}\ |\hat{\phi}^1_{i,j}(t)|>\bar{b}_2,\\ &\text{or}\ \text{sign}(\hat{\phi}^1_{i,j}(t))\neq\text{sign}(\hat{\phi}^1_{i,j}(1)),\ i\neq j\end{aligned} \tag{4.42}$$

$$u(t) =u(t-1)+\frac{\rho_1\hat{\phi}^T_1(t)(\bar{r}(t+1)-y(t))}{\tilde{\lambda}+\|\hat{\phi}_1(t)\|^2}. \tag{4.43}$$

MIMO PFDL-MFAC

$$\begin{aligned}\hat{\Phi}(t) =&\hat{\Phi}(t-1)\\ &+\frac{\eta(\Delta y(t)-\hat{\Phi}^T(t-1)\Delta\tilde{\mathbf{U}}(t-1))\Delta\tilde{\mathbf{U}}^T(t-1)}{\mu+\|\Delta\tilde{\mathbf{U}}(t-1)\|^2}\end{aligned} \tag{4.44}$$

$$\begin{aligned}\hat{\phi}^1_{i,i}(t) = \hat{\phi}^1_{i,i}(1),\ &\text{if}\ |\hat{\phi}^1_{i,i}(t)|<\bar{b}_1,\ \text{or}\ |\hat{\phi}^1_{i,i}(t)|>\alpha\bar{b}_1,\\ &\text{or}\ \text{sign}(\hat{\phi}^1_{i,i}(t))\neq\text{sign}(\hat{\phi}^1_{i,i}(1))\end{aligned} \tag{4.45}$$

$$\begin{aligned}\hat{\phi}^1_{i,j}(t) = \hat{\phi}^1_{i,j}(1),\ &\text{if}\ |\hat{\phi}^1_{i,j}(t)|>\bar{b}_2,\\ &\text{or}\ \text{sign}(\hat{\phi}^1_{i,j}(t))\neq\text{sign}(\hat{\phi}^1_{i,j}(1)),\ i\neq j\end{aligned} \tag{4.46}$$

$$\begin{aligned}u(t) =&u(t-1)+\frac{\rho_1\hat{\phi}^T_1(t)(\bar{r}(t+1)-y(t))}{\tilde{\lambda}+\|\hat{\phi}_1(t)\|^2}\\ &-\frac{\hat{\phi}^T_1(t)\sum_{k=2}^{L}\rho_k\hat{\phi}_k(t)\Delta u(t-k+1)}{\tilde{\lambda}+\|\hat{\phi}_1(t)\|^2}.\end{aligned} \tag{4.47}$$

4.3.5 Numerical example

To show some basic characteristics of MFAC protocol, a numerical example [30] is given as follows. Consider the plant which can be described by

$$\begin{aligned} y_1(t+1) =& \frac{2.5y_1(t)y_1(t-1)}{1+|y_1(t)|^2+|y_2(t-1)|^2+|y_2(t-2)|^2} \\ &+ 0.09u_1(t)u_1(t-1) + 1.2u_1(t) + 1.6u_1(t-2) + 0.5u_2(t) \\ &+ 0.7\sin(0.5(y_1(t)+y_1(t-1))) \cdot \cos(0.5(y_1(t)+y_1(t-1))) \end{aligned} \tag{4.48}$$

$$\begin{aligned} y_2(t+1) =& \frac{5y_2(t)y_2(t-1)}{1+|y_2(t)|^2+|y_1(t-1)|^2+|y_2(t-2)|^2} \\ &+ u_2(t) + 1.1u_2(t-1) + 1.4u_2(t-2) + 0.5u_1(t). \end{aligned} \tag{4.49}$$

The reference signal is set as

$$\bar{r}_1(t+1) = 5\sin(t/50) + 2\cos(t/20) \tag{4.50}$$

$$\bar{r}_2(t+1) = 2\sin(t/50) + 5\cos(t/20). \tag{4.51}$$

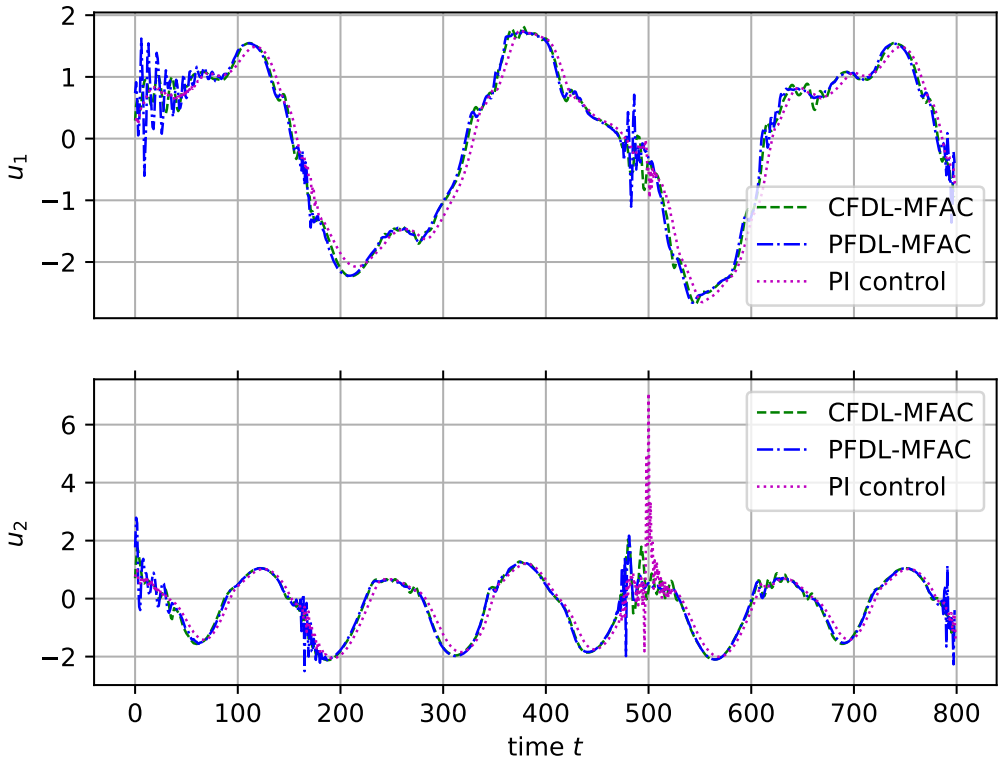


Figure 4.1: Numerical example: Inputs of CFDL-MFAC, PFDL-MFAC and PI control.

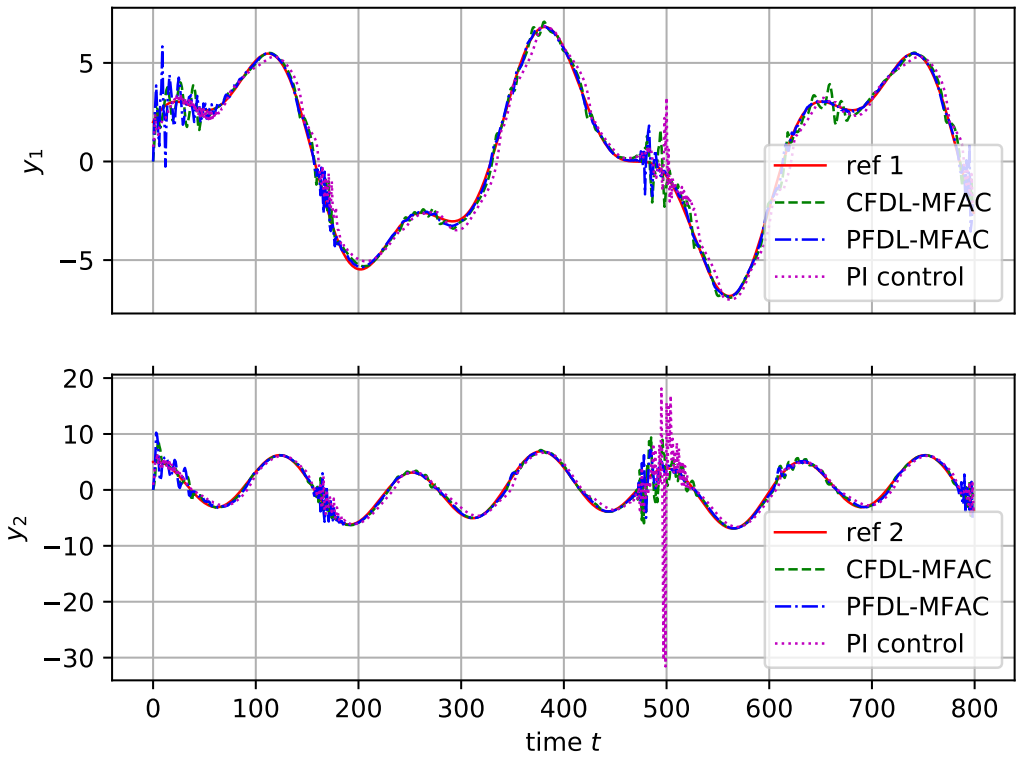


Figure 4.2: Numerical example: Outputs of CFDL-MFAC, PFDL-MFAC and PI control.

The tuning parameters of PI control are $K_p = 0.1$ and $K_i = 0.01$. The common tuning parameters of both CFDL- and PFDL-MFAC are $\mu = 1$, $\tilde{\lambda} = 0.01$, η, $\bar{b}_1 = 0.1$, and $\bar{b}_2 = 0.1$. The initial $\hat{\Phi}(t = 1)$ matrices of CFDL- and PFDL-MFAC are

$$\hat{\Phi}(t = 1) = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} \tag{4.52}$$

and

$$\hat{\Phi}(t = 1) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.53}$$

In this example, the performances of CFDL-MFAC, PFDL-MFAC and PI control protocols are demonstrated. As shown in Figure 4.1 and Figure 4.2, all these protocols can track time-varying reference signals. However, since MFAC includes parameterised estimation of dynamics for the plant, it is generally better than PI control. Moreover, since PFDL uses more parameters for model estimation than CFDL, the corresponding performance is the best in this example. The dynamic variations of the online model

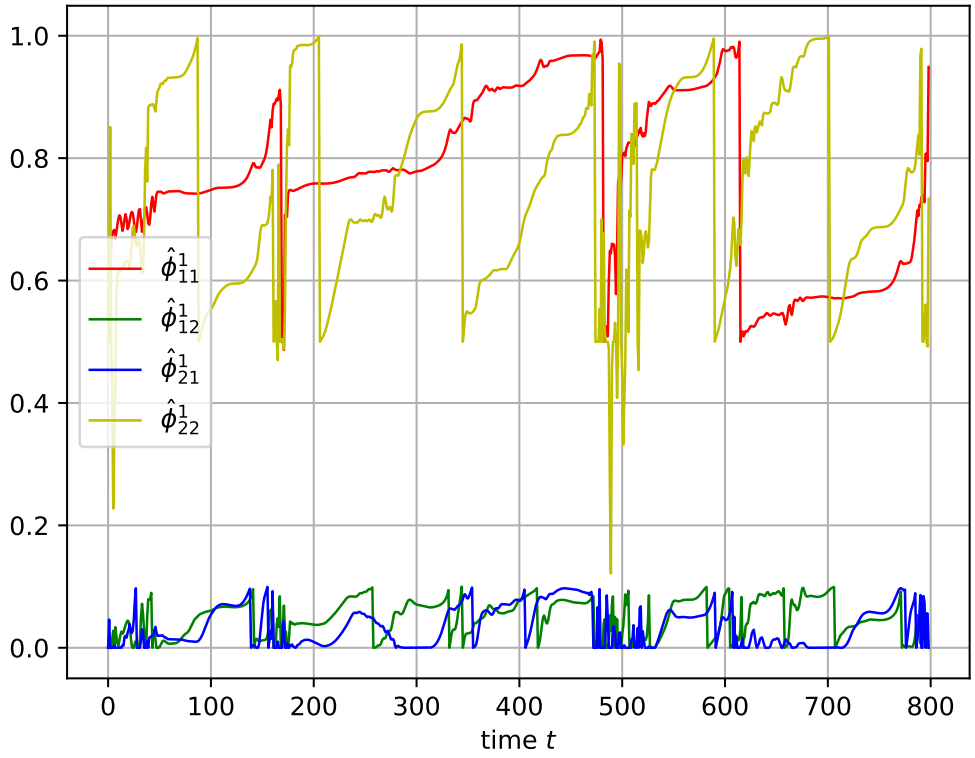


Figure 4.3: Numerical example: $\hat{\Phi}(t)$ of CFDL-MFAC.

parameters, i.e., the PPD matrices for CFDL- and PFDL-MFAC (Figures 4.3 and 4.4) are illustrated as well.

4.4 Simulation results

4.4.1 Comparison between MFAC and PI control

In this section, the performances of CFDL- and PFDL-MFAC are evaluated based on the previous validated neural-network-based PCC plant model,termed as the controlled plant in the subsequent sections. PI control results are also given for comparison. The lean MEA flow rate is the manipulated input while the carbon capture level is the controlled output. The original controlled plant is supposed to be free of disturbances. During the tuning process, K_p and K_i (Table 4.1) of PI control [43] are tuned to ensure tracking performances of the capture level as good as possible. Then, instead

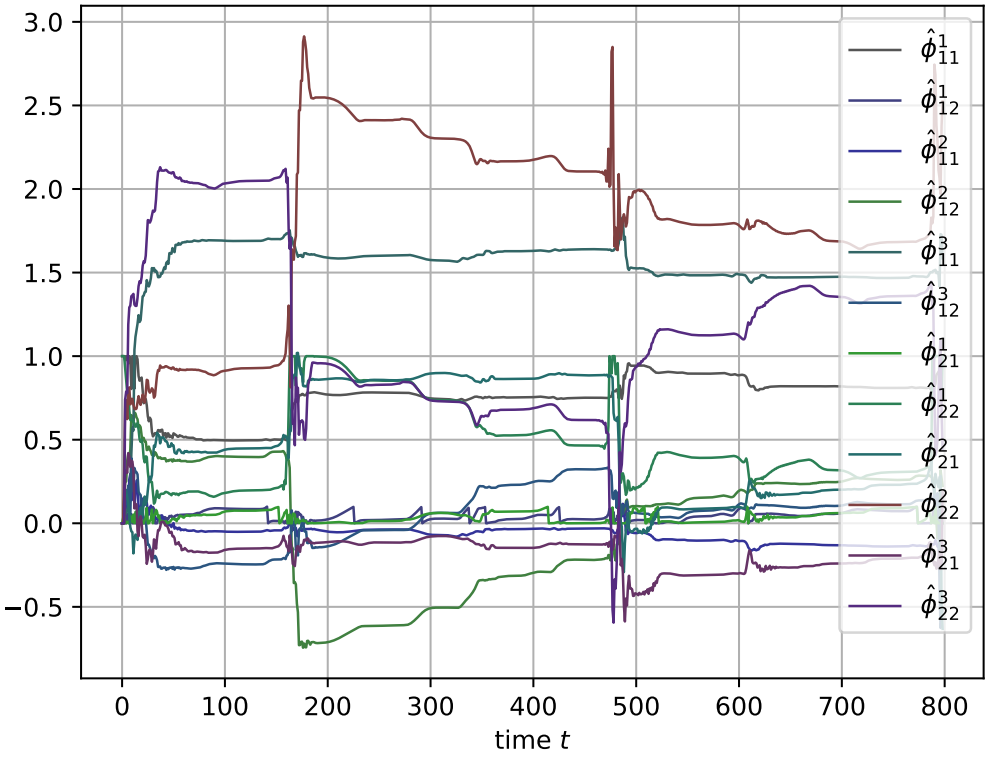


Figure 4.4: Numerical example: $\hat{\Phi}(t)$ of PFDL-MFAC.

of PI control, MFAC can be tuned as discussed in Subsection 4.3.2 and implemented to achieve similar performances (Figures 4.5 and 4.6) with the designed tuning parameters (Table 4.1). Although the number of tuning parameters for MFAC is larger than that for PI control, MFAC is easy to ensure stability [31]. PI control needs extra stability analysis of the closed-loop system.

Table 4.1: Controller design.

	PI		CFDL-MFAC	PFDL-MFAC
K_p	0.01	μ	0.002	0.002
K_i	0.017	$\tilde{\lambda}$	25	40
		$\boldsymbol{\rho}$	(1)	$(0.8, 0.05, 0.001)^T$
		α	200	200
		η	0.4	0.4
		b	0.1	0.1
		L	1	3
		$\hat{\Phi}(1)$	(3)	$(3, -5, -2)$

Afterwards, the time-variant disturbances, i.e., the flue gas flow rate and the CO_2

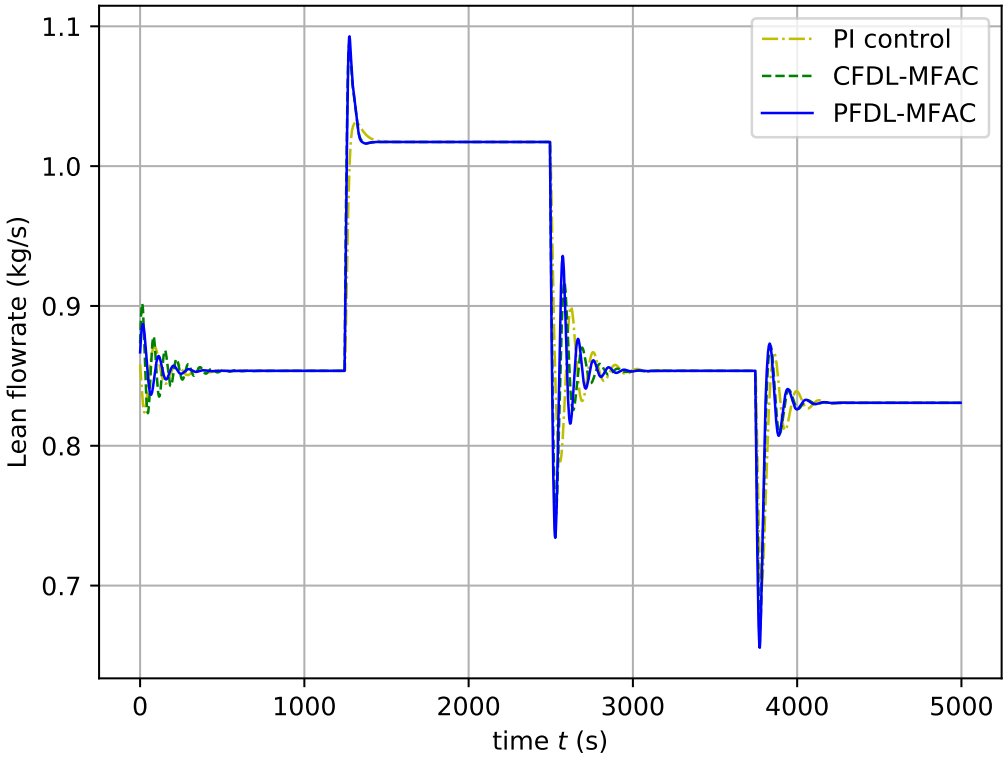


Figure 4.5: Manipulated inputs of different controllers with no disturbances applied.

mass fraction of the flue gas (Figure 3.13), are applied to the controlled nonlinear PCC plant, which can be periodical ramp changes due to the variations of power generation [14]. Such external disturbances cause time-variant dynamics of the controlled plant. Note that no compensation methods are included in the control protocols (i.e., MFAC or PI) so that only the robustness of different controllers are compared when confronting these perturbations. Simultaneously, the reference signal of the carbon capture level is generated identically to the one of the undisturbed plant (Figure 4.6). Based on the previous tuning parameters (Table 4.1), the lean-MEA flow rate (Figure 4.7), the capture level (Figure 4.8), and the capture level deviated from the references (Figure 4.9) are plotted, where PFDL-MFAC has the smoothest transient responses of the output, i.e. a smaller carbon capture level deviations than the PI control and CFDL-MFAC algorithms. PFDL-MFAC (Figure 4.8) is better than CFDL, since time-variant PPD $\hat{\Phi}(t)$ of PFDL with a longer length $L = 3$ (Table 4.1) adaptively catches more plant dynamics. CFDL-MFAC with fewer tuning parameters than PFDL-MFAC, however, can be designed more easily for simple plants [31].

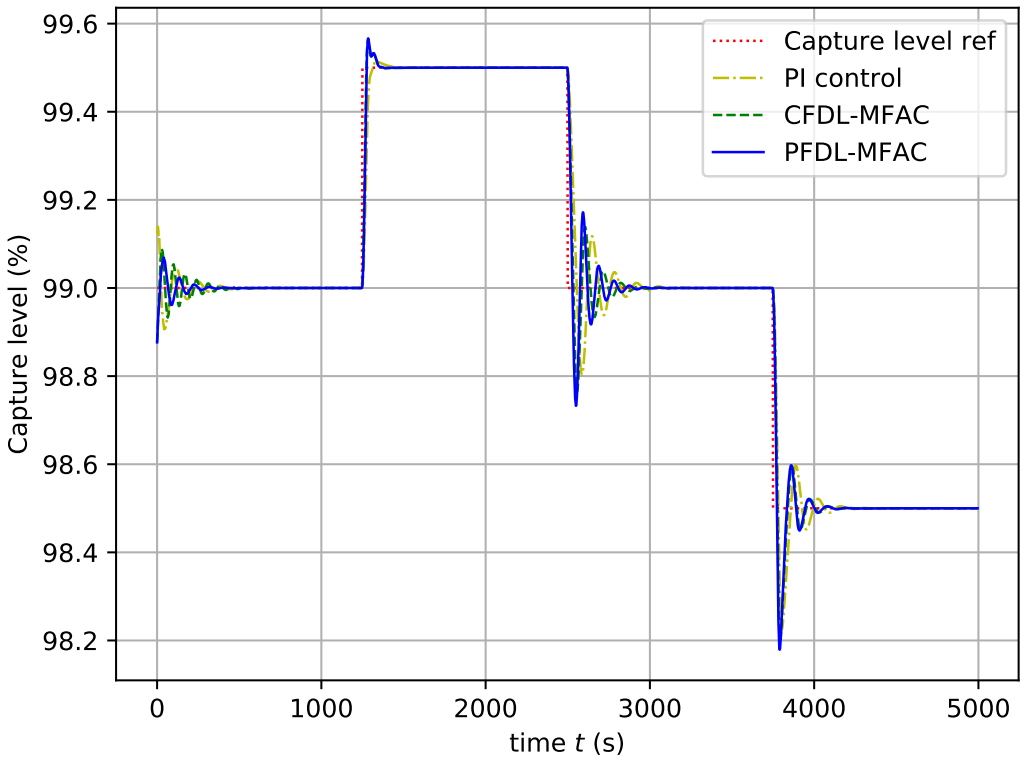


Figure 4.6: Controlled outputs of different controllers with no disturbances applied.

Both CFDL- and PFDL-MFAC can guarantee stability by increasing the stabilising parameter $\tilde{\lambda}$. Time-variant PPD vectors $\hat{\Phi}(t)$ due to CFDL and PFDL (Figures 4.10 and 4.11) dynamically estimate the controlled neural-network-based PCC plant.

4.4.2 Comparison between PFDL-MFAC and GPC

The tuning parameters of GPC are identical to those specified in Subsection 3.4.2. For convenience, they are presented as follows: $N_r = 3$, $\mathbf{Q} = 1 \cdot I_{N_r \times N_r}$ and $\mathbf{R} = 30 \cdot I_{N_r \times N_r}$ where $I_{N_r \times N_r} \in \mathbb{R}^{N_r \times N_r}$ is an identity matrix. PFDL-MFAC as tuned in Table 4.1 can achieve a similar tracking performance as GPC. Nevertheless, a prediction model should be identified before the tuning parameters of GPC can be tested online. The model not only lacks nonlinearities of the controlled plant but is usually obtained with offline sensitivity or identification tests. Both of them make the tuning procedure more complex than MFAC. The computation time of CFDL-MFAC, PFDL-MFAC, and GPC are 0.147 ms/sample, 0.189 ms/sample, and 0.194 ms/sample, respectively,

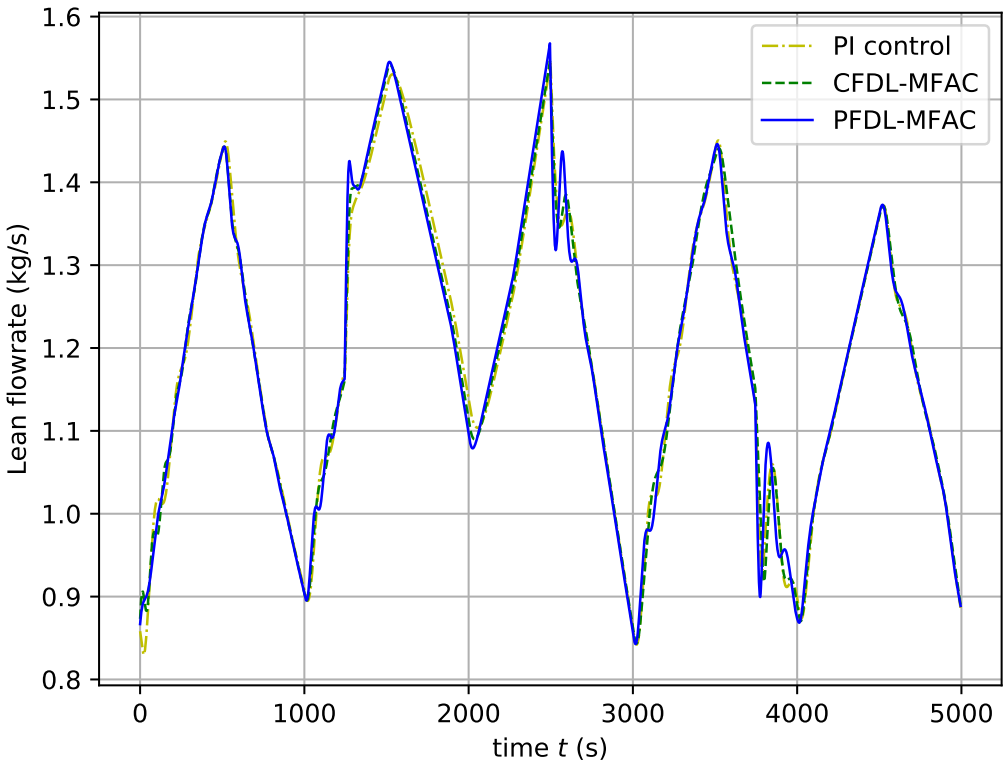


Figure 4.7: Manipulated inputs of different controllers with disturbances applied.

for the tuning parameters shown in Table 4.1, among which GPC is a little more time-consuming than the other two MFAC protocols. Nevertheless, the computation time for all these protocols are significantly less than the sampling time (2.5 s) or the rise time of the plant dynamics (about 80 s shown in Figure 3.12), which makes the control protocols feasible for the carbon capture level tracking issue.

4.5 Summary

In this chapter, some model-based and model-free controllers are implemented on a MEA-based PCC plant model, which shows that PFDL-MFAC is a robust and simple protocol for the tracking issue of the carbon capture level.

- We have implemented the PI control and the model-free algorithms, namely, CFDL- or PFDL-MFAC within the validated nonlinear PCC plant model. PFDL-MFAC has shown the best performance when confronting model uncertainties

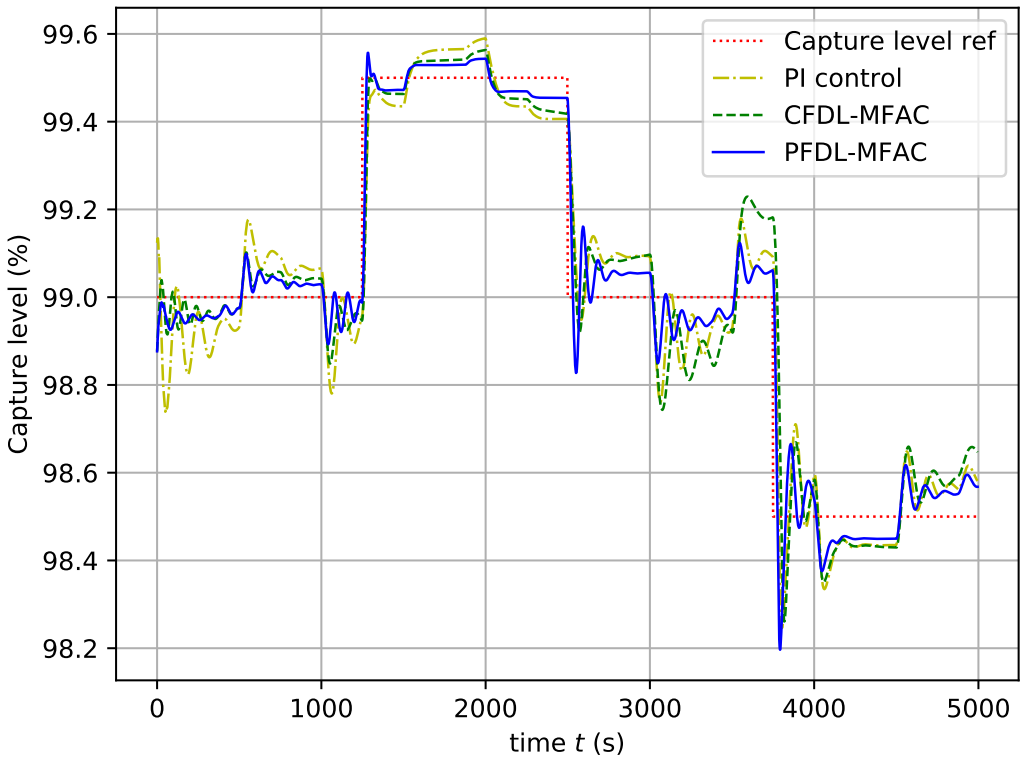


Figure 4.8: Controlled outputs of different controllers with disturbances applied.

caused by time-variant disturbances. CFDL-MFAC, however, can be tuned easily since it has fewer tuning parameters. Both CFDL- and PFDL-MFAC can guarantee the stability of the closed-loop system by the stabilising parameter $\tilde{\lambda}$, easier than PI control using the model-based Nyquist criterion.

- We have compared PFDL-MFAC with a model-based method called GPC. PFDL-MFAC can be more flexibly tuned online without model parameters determined during the offline system identification. GPC, however, must be applied based on underline models, which is linearised around specified equilibrium points of the controlled nonlinear plant. Extra time should be taken to ensure the model performances. When performances of such a model-based controller are unsatisfactory, re-identification of underline models may be required, which is non-trivial. Consequently, PFDL-MFAC can be flexibly designed and implemented easily online with a simplified offline tuning procedure.

- In this chapter, a novel MFAC protocol [30, 31] is applied to the neural-network-based MEA-based PCC plant model identified using the validated data generated

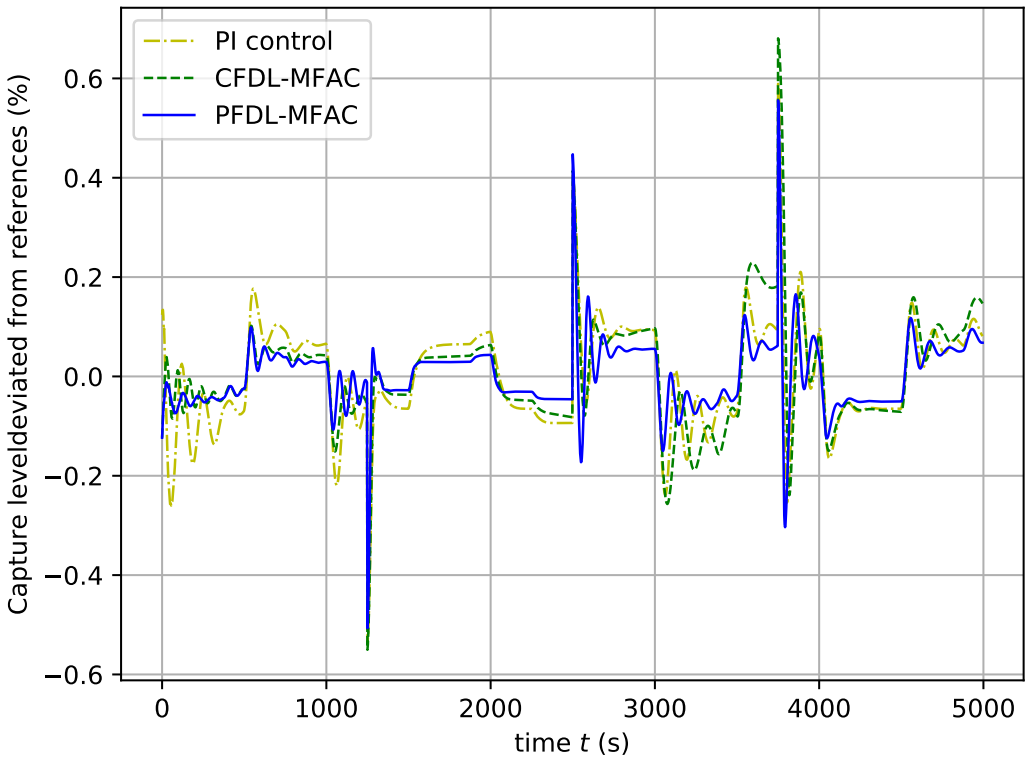


Figure 4.9: Controlled outputs deviated from references of different controllers with disturbances applied.

by a first-principle model [43]. Compared with PI control using predefined tuning parameters around fixed operating points, MFAC uses CFDL or PFDL to form a time-variant PCC model online, inferring that the model adapts to plant operating point changes. Compared with the model-based protocol which requires non-trivial sensitivity or identification tests to determine a model for offline tuning before online implementation, MFAC has a simpler tuning procedure. The identified PCC model is only used for the initial offline tuning. Thereafter, the tuning parameters can be flexibly retuned online with the measured input-output data of the controlled nonlinear PCC plant. No model parameters identified offline are required online. The model parameters determined by the transfer operators $\mathbf{A}(q)$, $\mathbf{B}(q)$, $\mathbf{C}(q)$ and $\mathbf{D}(q)$, however, are essential for GPC protocols. They are used to ensure the stability and performances of the closed-loop system, inferring a complex and repetitive offline tuning procedure. PI control requires no underline model parameters same as MFAC, but its stability analysis is based on models. MFAC can easily guarantee stability by a stabilising parameter

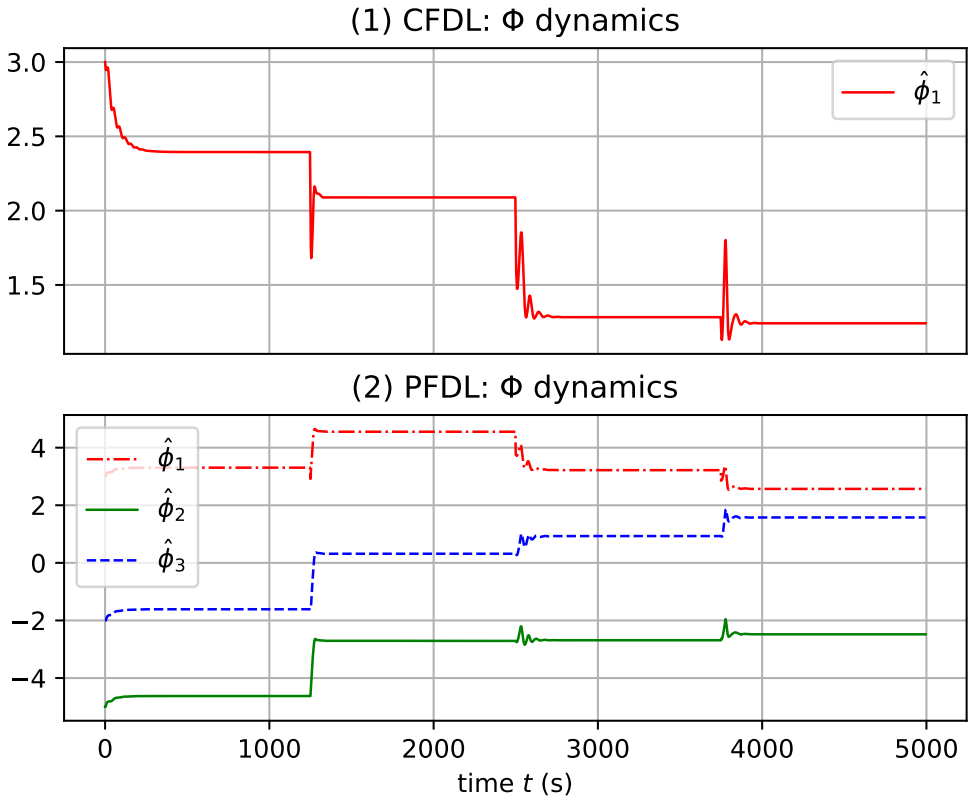


Figure 4.10: PPD vectors of CFDL- and PFDL-MFAC with no disturbances applied.

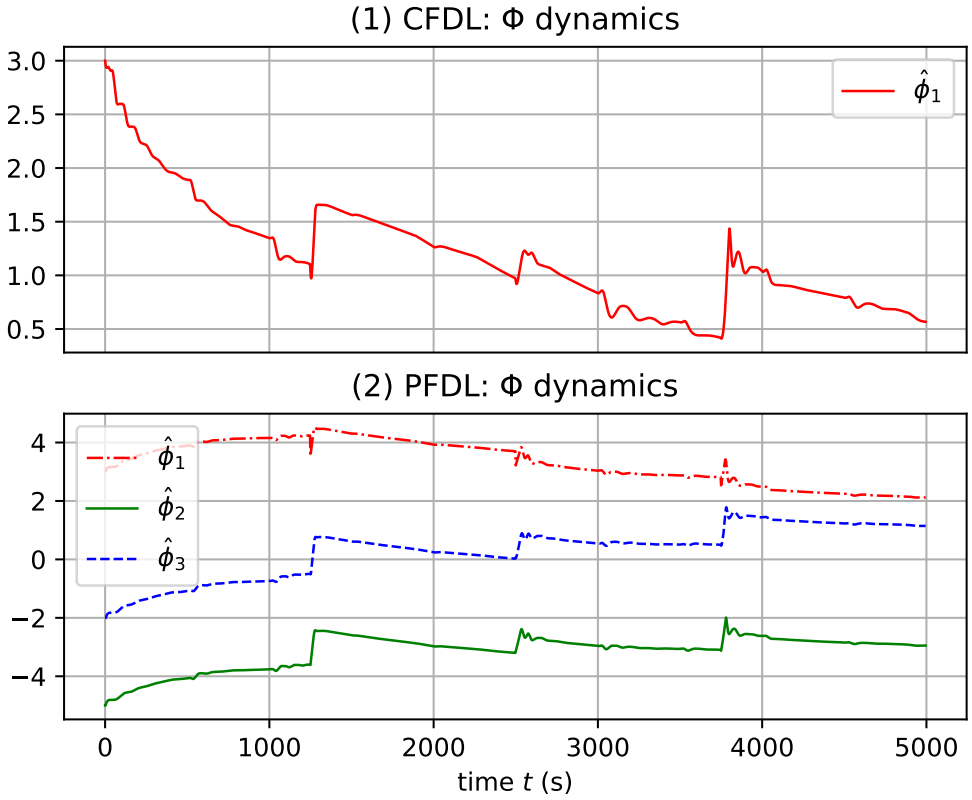


Figure 4.11: PPD vectors of CFDL- and PFDL-MFAC with disturbances applied.

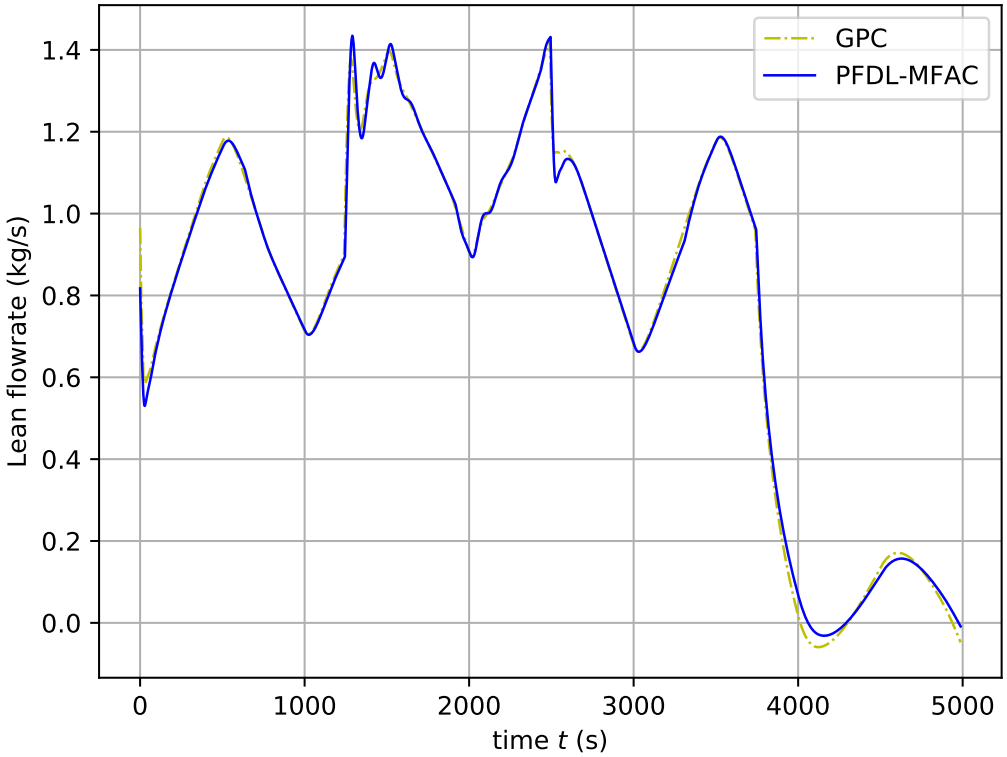


Figure 4.12: Manipulated inputs of MFAC-PFDL and GPC with disturbances applied.

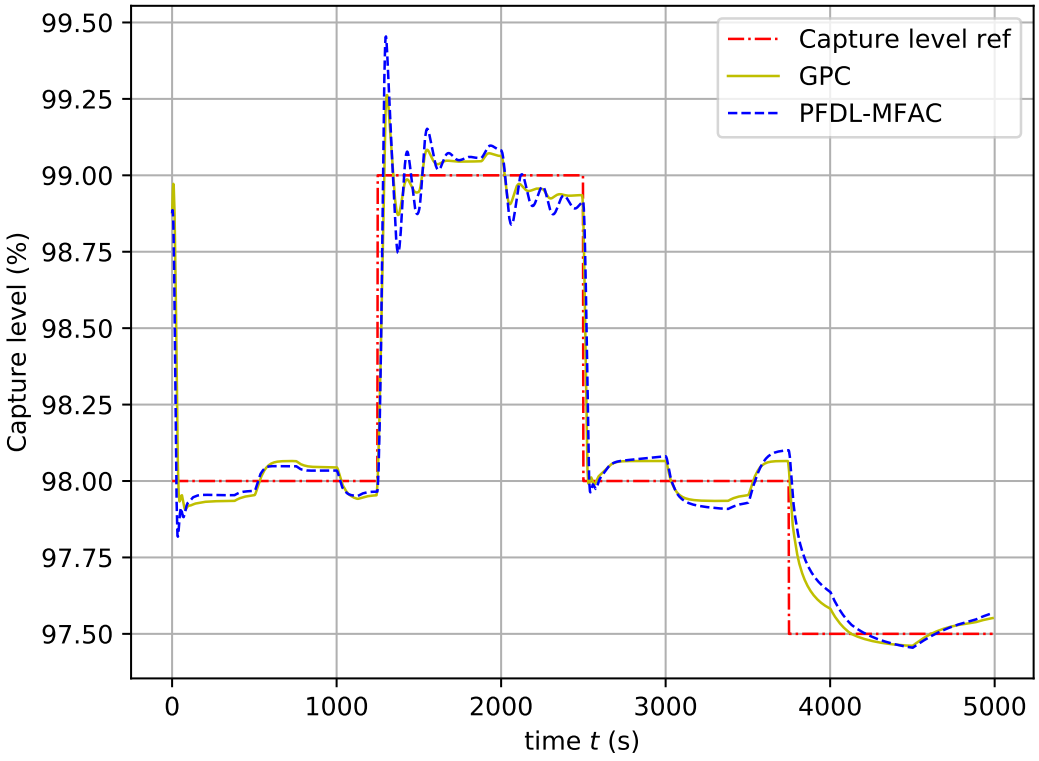


Figure 4.13: Controlled outputs of MFAC-PFDL and GPC with disturbances applied.

Chapter 5

Reinforcement-learning-based decision making of MEA-based post-combustion carbon capture process under emission trading system

5.1 Introduction

In previous chapters, we discussed the flexible control issues of an MEA-based PCC plant integrated with a fossil-fuel power plant. The tracking problem of the carbon capture level for the plant is solved using model-based (Chapter 3) and model-free (Chapter 4) controllers. However, from a higher-level perspective, the carbon capture level should be determined in some optimal route rather than some constant value. As mentioned in Section 1.2, the carbon capture level should be considered together with the CO_2 procurement through auctions under the emission trading system. It infers, to achieve optimal operation, the MEA-based PCC plant should make a unified decision on the carbon capture level and the bid option based on different electricity

and CO_2 allowance prices [6, 15, 17, 35, 40]. Although the carbon tax mechanism has been discussed in the aforementioned literature, the settlement price of the CO_2 allowances taking into account the auction mechanism is seldom investigated. Although there exist some literature talking about CO_2 allowance auctions [71] and bidding strategies [72], the flexible operation of the MEA-based PCC plant integrated with the fossil-fuel power plant is ignored. The relationship between PCC operation and allowance bidding is not fully established. Therefore, we tend to achieve the following targets. Firstly, the MEA-based PCC plant integrated with a fossil-fuel power plant should consider a profit-maximised objective with a flexible CO_2 capture level. Secondly, the MEA-based PCC plant should bid CO_2 allowances with appropriate quantities and prices. Note that one allowance is the permit of one metric ton of CO_2 emission. The winning CO_2 allowances from the allowance auctions will be consumed based on the CO_2 emission of the power plant. Inadequate procurement of CO_2 allowances will cause penalties based on the amount of the excessive greenhouse gas emission.

The above problem can be formulated as the Markov decision process and then solved with dynamic programming. However, the classic dynamic programming methods require perfect models and lots of computation resources. The new estimate of any state value must be achieved based on the previously estimated values of the whole state set. If the state set is large, both the computation time and memory usage will be significant. On the other hand, reinforcement learning methods usually require no models and their updates involve only the visited state-action pairs. These make reinforcement learning methods much more feasible. Sarsa is a simple on-policy reinforcement learning method. It can be implemented easily for the carbon capture decision-making problem above. As an extension of Sarsa, Sarsa(λ) is considered since it converges faster and can deal with delayed rewards/penalties. Note that there may exist some off-policy methods (e.g., Q-learning) which can be applied as well. However, such kinds of methods usually have slower convergence properties and larger variances. Moreover, even the combination of some simple linear function approximators and the off-policy learning will cause divergence problems [37]. Thus, we mainly focus on the Sarsa and Sarsa(λ) learning methods in this thesis.

5.2 Preliminaries

The operation and bidding problem of the MEA-based PCC plant will be solved under the reinforcement learning framework [36, 37, 84] whose theory is strongly dependent on dynamic programming [85, 86]. Therefore, in this section, some basics about dynamic programming and reinforcement learning are given for a better understanding of works in the subsequent sections.

5.2.1 Dynamic Programming

Dynamic programming [85, 86] solves a set of problems where the Markov decision process (MDP) and agent interact as follows: At time t, the agent can observe (or visit) a process state S_t, based on which it takes an action A_t under a predefined policy $\pi(\cdot|S_t)$ where $\pi(A_t|S_t)$ is the probability of taking action A_t conditional on state S_t. A reward R_{t+1} is then received by the agent while a new state S_{t+1} is observed. Recursively, the interactions between the agent and the environment give rise to a sequence

$$S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}, R_{t+2}, S_{t+2}, A_{t+2}, R_{t+3}, \cdots, A_{T-1}, R_T, S_T \tag{5.1}$$

with $S_k \in \mathcal{S}$, $A_k \in \mathcal{A}$, $R_{k+1} \in \mathcal{R}$ for $k \in \{t, t+1, \cdots, T-1\}$ where T is the terminal time. Provided $T < \infty$, such a sequence is termed as an episode. $\mathcal{S}$, $\mathcal{S}^+$, $\mathcal{A}$, and $\mathcal{R}$ are time-invariant finite sets for non-terminal states, states including terminals, actions, and rewards, respectively, for the preceding agent. Accordingly, it can be inferred that the terminal state S_T should satisfy $S_T \in \mathcal{S} \cap \mathcal{S}^+$. In what follows, S_k, A_k, R_{k+1} and S_T are random variables w.r.t. the possible values $s_k \in \mathcal{S}$, $a_k \in \mathcal{A}$, $r_{k+1} \in \mathcal{R}$, and $s_T \in \mathcal{S} \cap \mathcal{S}^+$. Additionally, any reward should be bounded by

$$|r_{k+1}| < \bar{b}_{\mathrm{r}} \tag{5.2}$$

with $\bar{b}_{\mathrm{r}} > 0$ and $\forall r_{k+1} \in \mathcal{R}$. Under the reinforcement learning framework, in this thesis, we seldom talk about the correlation between R_{k+1} and S_k. Thus, for a general problem, we usually express the joint probability of $R_{k+1} = r$ and $S_{k+1} = s'$ given the current visited state-action pair, (i.e., $S_k = s$ and $A_k = a$) as $p(s', r|s, a)$ [37]

constrained by

$$\sum_{s',r} p(s', r|s, a) \triangleq \sum_{s'} \sum_{r} p(s', r|s, a) \triangleq \sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1. \tag{5.3}$$

Given the sequence due to interaction between the agent and the environment, a discounted return is defined as

$$G_t \triangleq \sum_{k=t}^{T-1} \gamma^{k-t} R_{k+1} \tag{5.4}$$

where $0 < \gamma \leq 1$ is the discount rate. There are two kinds of tasks w.r.t. the above return function (Eq. (5.4)). One is termed as the episodic task while the other is termed as continuing task. For an episodic task, there is one additional constraint applied to the time horizon T, which is $T < \infty$; for the continuing task, there are no naturally stop mechanisms, i.e., $T = \infty$, which implies γ should be explicitly less than 1 to make the return bounded. However, for the episodic task, a specific terminal state s_T can be seen as an absorbing state which means:

- Given any action a under state s_T, the probability of transiting to s_T itself is unity, i.e., $s_T = S_{T+1} = S_{T+2} = \cdots$.
- Given any action a under state s_T, the rewards are always zeros, i.e., $0 = R_{T+1} = R_{T+2} = \cdots$.

In addition, for Eq. (5.3), the absorbing state s_T implies

$$\sum_{r \in \mathcal{R}} p(s_T, r|s_T, a) = 1. \tag{5.5}$$

Thus, an episodic task can be seen as a continuing task as well. A unified notation for both kinds of tasks can be obtained by letting $T = \infty$. The state-value function $v_\pi(\cdot)$ for policy π is defined as the expectation of the return (Eq. (5.4)) over the infinite time horizon, i.e.,

$$v_\pi(S_t = s) \triangleq v_\pi(s) \triangleq \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_{k+1} | S_t = s], \forall s \in \mathcal{S} \tag{5.6}$$

where $v_\pi(s)$ is the state value for policy π of state s. For $0 < \gamma < 1$, due to the boundedness of reward R_{k+1} (Eq. (5.2)), it can be obtained that

$$\begin{aligned} |v_\pi(S_t = s)| =& |\mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_{k+1} | S_t = s]| \\ \leq& \sum_{k=t}^{\infty} \gamma^{k-t} \bar{b}_{\mathrm{r}} = \frac{\bar{b}_{\mathrm{r}}}{1-\gamma}, \forall \pi. \end{aligned} \tag{5.7}$$

Note that the objective is to maximise the expected return, which is meaningful only if $v_\pi(s)$ (Eq. (5.6)) is bounded. Here, it is assured by two constraints: $0 < \gamma < 1$ and inequality (Eq. (5.2)). Note that if the task is episodic, i.e., $T < \infty$, then we can have the discount rate $\gamma = 1$. A user may have $T = \infty$ or $\gamma = 1$ but not both [37]. Subject to these conditions, it is said $v_\pi(s)$ is well defined. A policy is γ-optimal if

$$v_{\pi^*}(s) \triangleq v_*(s) \triangleq \sup_\pi v_\pi(s) \tag{5.8}$$

where $v_*(s)$ is the optimal state value of state s. From Eq. (5.6), it can be derived that

$$\begin{aligned} v_\pi(s) =& \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_{k+1} | S_t = s] \\ =& \sum_a \pi(a|s) \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_{k+1} | S_t = s, A_t = a] \\ =& \sum_a \pi(a|s) \mathbb{E}_\pi[R_t + \gamma \sum_{k=t+1}^{\infty} \gamma^{k-t-1} R_{k+1} | S_t = s, A_t = a] \\ =& \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)(r + \gamma \mathbb{E}_\pi[\sum_{k=t+1}^{\infty} \gamma^{k-t-1} R_{k+1} | S_t = s, A_t = a, S_{t+1} = s']) \\ & \text{(Since } R_{k+1} \text{ above is only related to } S_{t+1}, S_{t+2}, ..., \text{ we have)} \\ =& \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)(r + \gamma \mathbb{E}_\pi[\sum_{k=t+1}^{\infty} \gamma^{k-t-1} R_{k+1} | S_{t+1} = s']) \\ =& \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)(r + \gamma \mathbb{E}_\pi[\sum_{k=t+1}^{\infty} \gamma^{k} R_{k+2} | S_{t+1} = s']) \\ & \text{(Since Eq. (5.6) holds, we have)} \\ =& \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)(r + \gamma v_\pi(S_{t+1} = s')) \\ =& \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')] \end{aligned} \tag{5.9}$$

from which, together with Eq. (5.8), it can be proved that the optimal state value of any state s, or $v_*(s)$, should suffice the optimality equation [85, Theorem 2.1],

$$v_*(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')]. \tag{5.10}$$

Furthermore, $v_*(s)$ is the unique bounded solution of the optimality equation (Eq. (5.10)) [85, Proposition 2.3]. It should be noted that though the optimal state value $v_*(s)$ is unique, there may exist multiple optimal policies. Or precisely speaking, there exist at least one deterministic and stationary optimal policy.

A method termed as successive approximation can be used as follows to search an optimal policy: Specify the arbitrary initial guess of $v_*(s)$ through $V_0(s)$ for any non-terminal state $s \in \mathcal{S}$ and let $V_0(s) = 0$ for those terminal states $s \in \mathcal{S} \cap \mathcal{S}^+$. The successive approximations of $v_*(s)$ can be defined as $V_1(s)$ and recursively $V_n(s)$, which are

$$V_1(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V_0(s')] \tag{5.11}$$

and

$$V_n(s) = \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma V_{n-1}(s')]. \tag{5.12}$$

With $|\mathcal{S}|$ as the size or cardinality of $\mathcal{S}$, the finite set $\mathcal{S}$ can be further expressed as

$$\mathcal{S} \triangleq \{\tilde{s}_1, \tilde{s}_2, \cdots, \tilde{s}_{|\mathcal{S}|}\} \tag{5.13}$$

where $\{\tilde{s}_i\}$ is any orderly enumeration of the finite set $\mathcal{S}$. The first-round updates are implemented by traversing elements in $\mathcal{S}$ as follows: A state $s = \tilde{s}_1$ is firstly visited and updated using Eq. (5.11) such that the approximate state value $V_1(\tilde{s}_1)$ is obtained from $V_0(s)$ for $s \in \mathcal{S}$. Afterwards, the next state $s = \tilde{s}_2$ is visited where the objective-maximised action (Eq. (5.11)) is taken so as to update the approximate state value $V_1(\tilde{s}_2)$ from $V_0(s)$ for $s \in \mathcal{S}$. The second-round updates of $V_2(s)$ from $V_1(s)$ for all $s \in \mathcal{S}$ will start unless the approximate state values $V_1(s)$ for all $s \in \mathcal{S}$ are visited and updated thoroughly. Generally, the successive approximation (Eq. (5.12)) for the nth-round starts once the $n-1$th round updates to all the states are finished. Such an update process may last infinite rounds. It can be proved that, given an arbitrary initial estimate $V_0(s)$ of the optimal state values $v_*(s)$ for $\forall s \in \mathcal{S}$, $V_n(s) \to v_*(s)$ as $n \to \infty$ [85, Proposition 3.1].

5.2.2 Sarsa TD control

Define the action-value function q_π as

$$q_\pi(S_t = s, A_t = a) \triangleq q_\pi(s, a) \triangleq \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_{k+1} | S_t = s, A_t = a] \tag{5.14}$$

for $\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}$ where $q_\pi(s, a)$ is the value for policy π of state-action pair (s, a). Compared with the state-value function (Eq. (5.6)), it indicates

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s, a). \tag{5.15}$$

For the Sarsa algorithm, any action value for the current visited state-action pair can be estimated by

$$Q_{t+1}(S_t, A_t) = Q_t(S_t, A_t) + \alpha_t(R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t)) \tag{5.16}$$

where $Q(S_t, A_t)$ is the estimate of the action value $q_\pi(S_t, A_t)$, α_t is the learning rate at time t, $A_t \sim \pi(\cdot|S_t)$, and $A_{t+1} \sim \pi(\cdot|S_{t+1})$. A ε-greedy policy π can be used for action selection at time t, which is as follows.

- With the probability of ε_t, choose any action from the action set $\mathcal{A}$ uniformly random.
- With the probability of $1 - \varepsilon_t$, choose the greedy action $a = \arg\max_b Q(S_t, b)$.

Thus, for the current visited state $S_t = s$ at time t, the probability of taking a specific action $A_t = a$ can be represented by

$$\pi(a|s) = \begin{cases} 1 - \varepsilon_t + \frac{\varepsilon_t}{|\mathcal{A}|}, & \text{for } a = \arg\max_{b \in \mathcal{A}} Q(s, b) \\ \frac{\varepsilon_t}{|\mathcal{A}|}, & \text{for } a \neq \arg\max_{b \in \mathcal{A}} Q(s, b) \end{cases}. \tag{5.17}$$

To ensure convergence, two more constraints should be satisfied for the scheduling of the learning rate α_t and the probability of non-greediness ε_t. First, a set of the indicator functions, $\mathcal{I}_{\cdot,\cdot}$, is defined by

$$\begin{cases} \mathcal{I}_{s,S_t} = 1, \text{ if } s = S_t \\ \mathcal{I}_{s,S_t} = 0, \text{ if } s \neq S_t \\ \mathcal{I}_{a,A_t} = 1, \text{ if } a = A_t \\ \mathcal{I}_{a,A_t} = 0, \text{ if } a \neq A_t \end{cases}. \tag{5.18}$$

On the ground of Eq. (5.18), the learning rate α_t should be separately scheduled for each state-action pair (s, a), which is

$$\alpha_t(s,a) \triangleq \mathcal{I}_{s,S_t}\mathcal{I}_{a,A_t}\tilde{\alpha}_t(s,a). \tag{5.19}$$

Therefore, the Sarsa algorithm (Eq. (5.16) can be equivalently written as

$$\begin{cases} Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)(R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t)) \\ \alpha_t(s,a) = \mathcal{I}_{s,S_t}\mathcal{I}_{a,A_t}\tilde{\alpha}_t(s,a) \end{cases} \tag{5.20}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. From Eq. (5.19), it can be derived that $\alpha_t(s,a) = 0$ for any $(s,a) \in \{(s,a)|s \in \mathcal{S}, a \in \mathcal{A}, (s,a) \neq (S_t, A_t)\}$. Thus, at any time t, only the action value of $(s,a) = (S_t, A_t)$ is updated via Eq.(5.20), which implies Eq. (5.20) is equivalent to Eq. (5.16). Based on Eq. (5.19), for any specific state-action pair (s,a), $\{\tilde{\alpha}_t(s,a)\}$ is a sequence that makes Eq. (5.21) and Eq. (5.22) hold, i.e.,

$$\sum_{t=0}^{\infty} \alpha_t(s,a) = \infty \tag{5.21}$$

$$\sum_{t=0}^{\infty} |\alpha_t(s,a)|^2 < \infty \tag{5.22}$$

subject to $0 \leq \alpha_t(s,a) < 1$.

The non-greedy probability ε_t should be scheduled independently for each state as well. From Singh et al. [87], $\varepsilon_t(s)$ is computed as follows

$$\mathbf{n}_t(s) = \mathbf{n}_{t-1}(s) + \mathcal{I}_{s,S_t} \tag{5.23}$$

$$\varepsilon_t(s) = \frac{c}{\mathbf{n}_t(s)} \tag{5.24}$$

with $0 < c < 1$. $\mathbf{n}_t(s)$ is the number of visits to a specific state $s \in \mathcal{S}$ till time t.

In terms of above iterations, the Sarsa temporal difference (TD) control with ε-greedy policy is presented in Table 5.1. Note that though the convergence is theoretically ensured by the constraints Eq. (5.21), Eq. (5.22), Eq. (5.23), and Eq. (5.24) [87], one may still use some other possible scheduling methods of the associated tuning parameters, $\varepsilon_t(s)$ and $\alpha_t(s,a)$, for a real problem. On the one hand, tuning parameters subject to those constraints may require considerable tuning efforts to obtain satisfactory results. On the other hand, most real environments to be interacted with by

Table 5.1: Sarsa TD control with ε-greedy policy.

Schedule step size α and the non-greedy probability ε
Initialise $Q(s,a)$ for $(s,a) \in \mathcal{S} \times \mathcal{A}$ arbitrarily
Initialise $Q(s,\cdot) = 0$ for $s \in \mathcal{S} \cap \mathcal{S}^+$
while true
Initialise S
Select A based on $\pi(\cdot\|S)$ (Eq. (5.17))
do
Take action A, observe new state S', and obtain reward R
Select A' based on $\pi(\cdot\|S')$ (Eq. (5.17))
$Q(S,A) \leftarrow Q(S,A) + \alpha(R + \gamma Q(S',A') - Q(S,A))$
$S \leftarrow S'$, $A \leftarrow A'$
until S is a terminal state

the agents are non-stationary, which means the optimal policies may vary from time to time. In this case, a constant learning rate $\alpha_t = \alpha$ can naturally put more weights on the recently observed samples than the past, which is often appreciated to deal with those non-stationary environmental features [37]. The initialisation condition of $Q(s,a)$ can be seen as the prior knowledge of the action values (Eq. (5.14) and gives some biases for the action value estimation. Good initial conditions can encourage exploration of actions in the action set in case of finding only suboptimal policies [37]. These biases will eventually decrease to zero for some proper step sizes, e.g., constant step sizes. For any bounded initial conditions, reinforcement learning methods can make the action values converge to their optimal theoretically. Thus, in this thesis, we simply set the initial $Q(s,a)$ to be zeros for each state-action pair during initialisation.

5.2.3 Sarsa(λ)

From Kushner et al. [88], if the variable X_t is updated as

$$X_{t+1} = X_t + \bar{\alpha}_t(U_t - X_t) \tag{5.25}$$

where $0 \leq \bar{\alpha}_t < 1$, $\sum_{t=0}^{\infty} \bar{\alpha}_t = \infty$, $\sum_{t=0}^{\infty} |\bar{\alpha}_t|^2 < \infty$, and U_t are bounded random variable with mean $\bar{U}$, then

$$X_t \to \bar{U}, \text{ as } t \to \infty, \text{ with probability of 1.} \tag{5.26}$$

Reinforcement learning algorithms coincide with the idea that designs U_t whose mean value converges the optimal state $v_*(s)$ or action value $q_*(s,a)$. For the previous Sarsa algorithm (Eq. (5.20)), U_t should be set separately for each state-action pair $(s,a) \in \mathcal{S} \times \mathcal{A}$, which is denoted by $U_t(s,a)$. At time t, Eq. (5.19) and Eq. (5.25) indicate

$$X_{t+1}(s,a) = X_t(s,a) + \alpha_t(s,a)(U_t(s,a) - X_t(s,a)). \tag{5.27}$$

Together with Eq. (5.20), Eq. (5.27) implies

$$U_t(s,a) = R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) \triangleq G_{t:t+1} \tag{5.28}$$

where S_{t+1} and A_{t+1} are random variables w.r.t. the state and action for the next time step. This is a bootstrapping guess (i.e., the new guess is achieved through some previous guesses) of $q_\pi(s,a)$. Scheduling of the learning rate and ε-greedy policy (i.e., Eqs. (5.21), (5.22), (5.23), and (5.24)) is needed to make this estimate converge q_* with the probability of 1. $G_{t:t+1}$ is called a 1-step return which updates the estimate of the expected return of previously visited state-action pair (s,a), i.e., $q_\pi(s,a)$, after transiting to the next visited pair (S_{t+1}, A_{t+1}) stochastically. As an extension, to update the action value $q_\pi(s,a)$ of any state-action pair for n-step ahead, the n-step return is defined as

$$G_{t:t+n} \triangleq R_{t+1} + \gamma^1 R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}). \tag{5.29}$$

On that basis, the λ-return is defined as

$$G_t^\lambda \triangleq (1-\lambda) \sum_{k=t}^{T-2} \lambda^{k-t} G_{t:k+1} + \lambda^{T-t-1} G_t. \tag{5.30}$$

Note that since $(1-\lambda)\sum_{k=t}^{T-2} \lambda^{k-t} + \lambda^{T-t-1} = 1$, G_t^λ above naturally puts a specific probability distribution on returns with different lookahead steps.

Forward view of Sarsa(λ)

The λ-return, G_t^λ, is used as the estimate of q_π by the following Sarsa(λ) learning algorithm. In contrast with the Sarsa algorithm (Eq. (5.20)), the update rule is

$$\begin{cases} Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)(G_t^\lambda - Q_t(S_t, A_t)) \\ \alpha_t(s,a) = \mathcal{I}_{s,S_t} \mathcal{I}_{a,A_t} \tilde{\alpha}_t(s,a) \end{cases}. \tag{5.31}$$

Since G_t^λ is computed by multi-step lookahead, this rule can be referred to as the forward view of the Sarsa(λ) algorithm. Define

$$\delta_t^\lambda \triangleq G_t^\lambda - Q_t(S_t, A_t) \tag{5.32}$$

$$\delta_t \triangleq R_{t+1} + \gamma Q_t(S_{t+1}, A_{t+1}) - Q_t(S_t, A_t). \tag{5.33}$$

Eq. (5.31) can be simplified as

$$Q_{t+1}(s,a) = Q_t(s,a) + \alpha_t(s,a)\delta_t^\lambda. \tag{5.34}$$

Together with Eq. (5.29) and Eq. (5.30), Eq. (5.32) can be rewritten as

$$\begin{aligned}
\delta_t^\lambda \triangleq & (1-\lambda)\sum_{k=t}^{T-2}\lambda^{k-t}G_{t:k+1} + \lambda^{T-t-1}G_t - Q_t(S_t, A_t) \\
= & (1-\lambda)\lambda^0(R_{t+1} + \gamma^1 Q_t(S_{t+1}, A_{t+1})) \\
& + (1-\lambda)\lambda^1(R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 Q_{t+1}(S_{t+2}, A_{t+2})) \\
& + (1-\lambda)\lambda^2(R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 R_{t+2} + \gamma^3 Q_{t+3}(S_{t+3}, A_{t+3})) \\
& + \cdots \\
& + (1-\lambda)\lambda^{T-t-2}(R_{t+1} + \gamma^1 R_{t+2} + \cdots + \gamma^{T-t-2}R_{T-1} + \gamma^{T-t-1}Q_{T-2}(S_{T-1}, A_{T-1})) \\
& + \lambda^{T-t-1}(R_{t+1} + \gamma^1 R_{t+2} + \cdots + \gamma^{T-t-1}R_T + \gamma^{T-t}Q_{T-1}(S_T, A_T)) \\
& - Q_t(S_t, A_t) \\
= & (\gamma^0\lambda^0 R_{t+1} + \gamma^1\lambda^0 Q_t(S_{t+1}, A_{t+1}) - \gamma^1\lambda^1 Q_t(S_{t+1}, A_{t+1})) \\
& + (\gamma^1\lambda^1 R_{t+2} + \gamma^2\lambda^1 Q_{t+1}(S_{t+2}, A_{t+2}) - \gamma^2\lambda^2 Q_{t+1}(S_{t+2}, A_{t+2})) \\
& + (\gamma^2\lambda^2 R_{t+3} + \gamma^3\lambda^2 Q_{t+2}(S_{t+3}, A_{t+3}) - \gamma^3\lambda^3 Q_{t+2}(S_{t+3}, A_{t+3})) \\
& + \cdots \\
& + (\gamma^{T-t-2}\lambda^{T-t-2}R_{T-1} + \gamma^{T-t-1}\lambda^{T-t-2}Q_{T-2}(S_{T-1}, A_{T-1}) \\
& \qquad - \gamma^{T-t-1}\lambda^{T-t-1}Q_{T-2}(S_{T-1}, A_{T-1})) \\
& + (\gamma^{T-t-1}\lambda^{T-t-1}R_T + \gamma^{T-t}\lambda^{T-t-1}Q_{T-1}(S_T, A_T)) \\
& - Q_t(S_t, A_t).
\end{aligned} \tag{5.35}$$

There is a causality problem in computing δ_t^λ online with Eq. (5.35). Thus, at any specific time t, we assume

$$Q_t(s,a) = Q_{t+1}(s,a) = \cdots = Q_{T-1}(s,a) \tag{5.36}$$

so as to derive δ_t^λ in an offline manner. Combining Eq. (5.35) and Eq. (5.36) gives

$$\begin{aligned}\delta_t^\lambda =&\gamma^0\lambda^0(R_{t+1}+\gamma Q_t(S_{t+1},A_{t+1})-Q_t(S_t,A_t))\\&+\gamma^1\lambda^1(R_{t+2}+\gamma Q_t(S_{t+2},A_{t+2})-Q_t(S_{t+1},A_{t+1}))\\&+\gamma^2\lambda^2(R_{t+3}+\gamma Q_t(S_{t+3},A_{t+3})-Q_t(S_{t+2},A_{t+2}))\\&+\cdots\\&+\gamma^{T-t-1}\lambda^{T-t-1}(R_T+\gamma Q_t(S_T,A_T)-Q_t(S_{T-1},A_{T-1})).\end{aligned}\tag{5.37}$$

In addition, the following approximation is used in comparison with Eq. (5.33),

$$R_{k+1}+\gamma Q_t(S_{k+1},A_{k+1})-Q_t(S_k,A_k)\approx\delta_k\tag{5.38}$$

for $k\in\{t+1,t+2,\cdots,T-1\}$. Eq. (5.37) can be briefly written as

$$\delta_t^\lambda\approx\sum_{k=t}^{T-1}(\gamma\lambda)^{k-t}\delta_k.\tag{5.39}$$

The total update of any specific state-action pair from time $t=0$ to $T-1$ in the forward view is defined as

$$\begin{aligned}\tilde{Q}_T^{\mathrm{f}}(s,a)\triangleq&Q_T(s,a)-Q_0(s,a)\\=&\sum_{t=0}^{T-1}\alpha_t(s,a)\delta_t^\lambda\end{aligned}\tag{5.40}$$

for all $s\in\mathcal{S}$ and $a\in\mathcal{A}$. With the above approximation (Eq. (5.39)), Eq. (5.40) can be transformed into

$$\begin{aligned}\tilde{Q}_T^{\mathrm{f}}(s,a)\approx&\sum_{t=0}^{T-1}\alpha_t(s,a)\sum_{k=t}^{T-1}(\gamma\lambda)^{k-t}\delta_k\\=&\sum_{t=0}^{T-1}\mathcal{I}_{s,S_t}\mathcal{I}_{a,A_t}\tilde{\alpha}_t(s,a)\sum_{k=t}^{T-1}(\gamma\lambda)^{k-t}\delta_k.\end{aligned}\tag{5.41}$$

Provided that $\tilde{\alpha}_t(s,a)=\tilde{\alpha}(s,a)$ for time $t\in\{0,1,\cdots,T-1\}$, Eq. (5.41) can be converted to

$$\tilde{Q}_T^{\mathrm{f}}(s,a)\approx\tilde{\alpha}(s,a)\sum_{t=0}^{T-1}\mathcal{I}_{s,S_t}\mathcal{I}_{a,A_t}\sum_{k=t}^{T-1}(\gamma\lambda)^{k-t}\delta_k.\tag{5.42}$$

Backward view of Sarsa(λ)

For the Sarsa(λ) algorithm in the backward view, the update rule is

$$\begin{cases} Q_{t+1}(s,a) = Q_t(s,a) + \tilde{\alpha}_t(s,a)\delta_t z_t(s,a) \\ z_t(s,a) = \gamma\lambda z_{t-1}(s,a) + \mathcal{I}_{s,S_t}\mathcal{I}_{a,A_t} \end{cases} \tag{5.43}$$

where $z_t(s,a)$ above is termed as the eligibility trace with $z_{-1}(s,a) = 0$ for the state-action pair (s,a). In contrast with Eq. (5.31), this algorithm can be implemented online without the preceding causality problem. Recursively, the eligibility trace of any state-action pair can be derived as follows

$$\begin{aligned} z_t(s,a) =& \mathcal{I}_{s,S_t}\mathcal{I}_{a,A_t} + (\gamma\lambda)^1\mathcal{I}_{s,S_{t-1}}\mathcal{I}_{a,A_{t-1}} + (\gamma\lambda)^2\mathcal{I}_{s,S_{t-2}}\mathcal{I}_{a,A_{t-2}} \\ &+ \cdots + (\gamma\lambda)^t\mathcal{I}_{s,S_0}\mathcal{I}_{a,A_0} \\ =& \sum_{k=0}^{t}(\gamma\lambda)^k\mathcal{I}_{s,S_{t-k}}\mathcal{I}_{a,A_{t-k}}. \end{aligned} \tag{5.44}$$

The above representation infers that eligibility traces record the number of historical visits for each state-action pair with time-related decays. Thus, it is termed as a backward-view method. With some assumptions, the backward-view algorithm (Eq. (5.43)) can be seen as the approximation of the forward-view Sarsa(λ) (Eq. (5.31)), which is presented as follows. The total updates of any specific state-action pair from time $t = 0$ to T can be written as

$$\begin{aligned} \tilde{Q}_T^{\mathrm{b}}(s,a) \triangleq& Q_T(s,a) - Q_0(s,a) \\ =& \sum_{t=0}^{T-1}\tilde{\alpha}_t(s,a)\delta_t z_t(s,a) \\ =& \sum_{t=0}^{T-1}\tilde{\alpha}_t(s,a)\delta_t \sum_{k=0}^{t}(\gamma\lambda)^k\mathcal{I}_{s,S_{t-k}}\mathcal{I}_{a,A_{t-k}} \end{aligned} \tag{5.45}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. With $m = t - k$, it derives

$$\begin{aligned} \tilde{Q}_T^{\mathrm{b}}(s,a) =& \sum_{t=0}^{T-1}\tilde{\alpha}_t(s,a)\delta_t \sum_{m=0}^{t}(\gamma\lambda)^{t-m}\mathcal{I}_{s,S_m}\mathcal{I}_{a,A_m} \\ =& \sum_{m=0}^{T-1}\sum_{t=m}^{T-1}(\gamma\lambda)^{t-m}\mathcal{I}_{s,S_m}\mathcal{I}_{a,A_m}\tilde{\alpha}_t(s,a)\delta_t \\ =& \sum_{m=0}^{T-1}\mathcal{I}_{s,S_m}\mathcal{I}_{a,A_m}\sum_{t=m}^{T-1}(\gamma\lambda)^{t-m}\tilde{\alpha}_t(s,a)\delta_t. \end{aligned} \tag{5.46}$$

For a constant learning rate during time $t \in \{0, 1, \cdots, t-1\}$, i.e., $\tilde{\alpha}_t(s,a) = \tilde{\alpha}(s,a)$, Eq. (5.46) can be written as

$$\tilde{Q}_T^{\mathrm{b}}(s,a) = \tilde{\alpha}(s,a) \sum_{m=0}^{T-1} \mathcal{I}_{s,S_m} \mathcal{I}_{a,A_m} \sum_{t=m}^{T-1} (\gamma\lambda)^{t-m} \delta_t. \tag{5.47}$$

Comparing Eq. (5.42) and Eq. (5.47) gives

$$\tilde{Q}_T^{\mathrm{b}}(s,a) \approx \tilde{Q}_T^{\mathrm{f}}(s,a). \tag{5.48}$$

It should be noted that if the Sarsa(λ) in the backward view (Eq. (5.43)) is updated offline, with the assumption that $\tilde{\alpha}_t(s,a) = \tilde{\alpha}(s,a)$, it is exactly same as the Sarsa(λ) in the forward view, i.e., $\tilde{Q}_T^{\mathrm{b}}(s,a) = \tilde{Q}_T^{\mathrm{f}}(s,a)$. Since the backward-view Sarsa(λ) can be applied online, it is more favourable. By default, without any other statement, Sarsa(λ) in the following sections means the preceding backward-view Sarsa(λ) learning algorithm. The online implementation of Sarsa(λ) is shown in Table 5.2.

Table 5.2: Sarsa(λ) control with ε-greedy policy.

Schedule step size α and the non-greedy probability ε
Initialise $Q(s,a)$ for $(s,a) \in \mathcal{S} \times \mathcal{A}$ arbitrarily
Initialise $Q(s,\cdot) = 0$ for $s \in \mathcal{S} \cap \mathcal{S}^+$
while true
  Initialise S
  Select A based on $\pi(\cdot|S)$ (Eq. (5.17))
  do
    Take action A, observe new state S', and obtain reward R
    Select A' based on $\pi(\cdot|S')$ (Eq. (5.17))
    $\delta \leftarrow R + \gamma Q(S', A') - Q(S, A)$
    $z(S,A) \leftarrow z(S,A) + 1$
    for each (s,a) in $\mathcal{S} \times \mathcal{A}$
      $Q(s,a) \leftarrow Q(s,a) + \alpha\delta z(s,a)$
      $z(s,a) \leftarrow \gamma\lambda z(s,a)$
    $S \leftarrow S'$, $A \leftarrow A'$
  until S is a terminal state

5.3 Problem formulation

In this section, the decision-making problem of a carbon capture plant integrated with a fossil-fuel power plant is formulated. The objective of the plant is to maximise the

profit considering the flexible carbon capture levels and bid options submitted to the CO_2 allowance auctions. A plant can achieve its emission target by procuring CO_2 allowances from auctions, directly reducing the emission through the MEA-based PCC technology, or both. Following assumptions are made to simplify the problem.

(i) Any MEA-based PCC plant is a price taker of the wholesale electricity market. Presumed electricity price and power generation profiles are given for all the plants (Figures 5.1 and 5.2).

(ii) Presumed CO_2 allowance reserve prices and scheduled allowance volumes are common knowledge among all the auction participants (Figures 5.3 and 5.4).

(iii) Any fossil-fuel power plant with carbon capture is retrofitted from a conventional fossil-fuel power plant without carbon capture. Hard coal and natural gas are possible power generation technologies.

(iv) There is no downsizing of any power plant facility after the integration of the carbon capture plant. The power plants can still operate at their nominal power outputs as before retrofitting when the power demands are high. When power generation and carbon capture conflict with each other, the power generation has a higher priority.

(v) The operation of either a power plant or the integrated carbon capture plant is characterised with some specific constant coefficients, i.e., $\mathcal{N}$, $\mathcal{E}$, $\mathcal{P}_{\text{cc}}$, $\mathcal{Q}_{\text{reb}}$, and $\mathcal{V}_{\text{cc}}$ (Tables 5.3 and 5.4).

- The specific non-fuel operation and maintenance (OM) cost (€/MWh) $\mathcal{N}$ is the OM cost (€) per unit of electricity production for a power plant without carbon capture.
- The specific emission (kg_{CO_2}/GJ) $\mathcal{E}$ is the CO_2 mass flow rate (kg_{CO_2}) per unit of fuel consumption (GJ) in lower heat value (LHV) for a power plant without carbon capture.
- The specific power consumption (MWh/t_{CO_2}) $\mathcal{P}_{\text{cc}}$ is the power consumptions (MWh) per unit of CO_2 capture (t_{CO_2}) for a carbon capture plant.

- The specific reboiler duty (GJ/t_{CO_2}) $\mathcal{Q}_{\text{reb}}$ is the reboiler energy consumptions (GJ) per unit of CO_2 capture (t_{CO_2}) for a carbon capture plant.
- The specific variable OM cost (€/t_{CO_2}) $\mathcal{V}_{\text{cc}}$ is the OM cost (€) per unit of CO_2 capture (t_{CO_2}) for a carbon capture plant.

Table 5.3 and Table 5.4 only give the base operation parameters of power plants and carbon capture plants separately. The trade-off between power generation and carbon capture will be introduced in following sections. On that basis, under the CO_2 allowance auctions, the operation and bidding problem of a set of competitive carbon capture plants integrated with fossil-fuel power plants are formulated.

Table 5.3: Specifications of power plants without carbon capture [1, 2].

Technology	Parameters	Symbol	Value
Coal	Nominal capacity (MW)	P_{gB}	650
	Capacity factor (%) [1]	ζ	49.29
	Efficiency (%) [1]	η	45
	Non-fuel OM cost (€/MWh) [1]	$\mathcal{N}$	7.35
	Coal price (€/GJ) [1]	f	3.68
	Lifetime (years) [1]	L_{w}	40
	CO_2 in flue gas (kg_{CO_2}/s) [2]	$\dot{m}_{\text{B}}$	132.63
	Specific emission (kg_{CO_2}/GJ) [2]	$\mathcal{E}$	91.82
Natural gas	Nominal capacity (MW)	P_{gB}	650
	Capacity factor (%) [1]	ζ	21.29
	Efficiency (%) [1]	η	55
	Non-fuel OM cost (€/MWh) [1]	$\mathcal{N}$	6.3
	Gas price (€/GJ) [1]	f	5.88
	Lifetime (years) [1]	L_{w}	30
	CO_2 in flue gas (kg_{CO_2}/s) [2]	$\dot{m}_{\text{B}}$	65.93
	Specific emission (kg_{CO_2}/GJ) [2]	$\mathcal{E}$	55.78

*An inflation rate of 5% is applied to those data recorded in [1].

Table 5.4: MEA-based PCC plants parameters for different power plants [3].

Technology	Parameters	Symbol	Value
Coal	CO_2 in flue gas (kg_{CO_2}/s)	$\dot{m}_{\text{B}}$	162.7
	Annual CO_2 emission ($\times 10^5 t_{CO_2}$)		39.25
	Capture level (%)		90
	PCC load (MW)		67.4
	Specific load (MWh/t_{CO_2})	$\mathcal{P}_{\text{cc}}$	0.128
	MEA regeneration energy (GJ/t_{CO_2})	$\mathcal{Q}_{\text{reb}}$	3.70
	Penalty coefficient		0.5
	Variable OM cost (€/t_{CO_2})	$\mathcal{V}_{\text{cc}}$	3.9
Natural Gas	CO_2 in flue gas (kg_{CO_2}/s)	$\dot{m}_{\text{B}}$	81.07
	Annual CO_2 emission ($\times 10^5 t_{CO_2}$)		19.56
	Capture level (%)		90
	PCC load (MW)		46.58
	Specific load (MWh/t_{CO_2})	$\mathcal{P}_{\text{cc}}$	0.177
	MEA regeneration energy (GJ/t_{CO_2})	$\mathcal{Q}_{\text{reb}}$	3.95
	Penalty coefficient		0.455
	Variable OM cost (€/t_{CO_2})	$\mathcal{V}_{\text{cc}}$	2.43

5.3.1 Economic model of a fossil-fuel power plant with carbon capture

In this section, the operational cost and profit for a fossil-fuel power plant with carbon capture for each auction period T_{a} (which is assumed to be two weeks or 336 hours = 24 hours $\times$ 14) are formulated.

The operational cost of a fossil-fuel power plant with carbon capture can be divided into the non-fuel OM cost for the power plant without carbon capture N_{p}, variable OM cost of the integrated carbon capture plant V_{cc}, the fuel cost F_{cc}, and the allowance bidding cost B, i.e.,

$$\begin{aligned} C_{i,t} \triangleq & N_{\mathrm{p}i,t} + V_{\mathrm{cc}i,t} + F_{i,t} + B_{i,t} \\ = & \mathcal{N}_i P_{\mathrm{gg}i,t} T_{\mathrm{a}} + \mathcal{V}_{\mathrm{cc}i} \frac{P_{\mathrm{gg}i,t}}{\eta_i} \mathcal{E}_i y_{i,t} T_{\mathrm{a}} + f \frac{P_{\mathrm{gg}i,t}}{\eta_i} T_{\mathrm{a}} + p_{\mathrm{st}} q_{\mathrm{s}i,t} \\ = & (\mathcal{N}_i + \frac{\mathcal{V}_{\mathrm{cc}i} \mathcal{E}_i y_{i,t} + f}{\eta_i}) T_{\mathrm{a}} P_{\mathrm{gg}i,t} + p_{\mathrm{st}} q_{\mathrm{s}i,t} \end{aligned} \tag{5.49}$$

where P_{gg} is the gross power output of the power plant, η is the thermal efficiency on the basis of the LHV, y is the capture level of CO_2 in the flue gas of the power plant, and f is the fuel price. The last term to the right of Eq. (5.49) is the bidding cost dependent on the winning bid quantity q_{s} of agent i and the unified settlement price p_{s} of all the agents. Some more details will be introduced in Subsection 5.3.2.

The gross power output P_{gg} can be either used for power generation of a power plant or aborted since some steam should be drawn off from the turbine for carbon capture. This kind of relationship can be built with

$$\begin{aligned} P_{\mathrm{gg}i,t} = & P_{\mathrm{g}i,t} + \mathcal{P}_{i,t} P_{\mathrm{gg}i,t} \\ & + \mathcal{P}_{\mathrm{cc}i} \mathcal{E}_i \frac{P_{\mathrm{gg}i,t}}{\eta_i} y_{i,t} + \psi_i \eta_i \mathcal{Q}_{\mathrm{reb}i} \frac{P_{\mathrm{gg}i,t}}{P_{\mathrm{gB}i}} \dot{m}_{\mathrm{B}i} y_{i,t} \end{aligned} \tag{5.50}$$

where the first and second terms to the right hand side of Eq. (5.50) are the in-house power load and the net power output, respectively, of the power plant without carbon capture; the third and fourth terms are the in-house load and the power penalty due to carbon capture, respectively, of the carbon capture plant. From Eq. (5.50), there exists some trade-off between the power generation and the energy-intensive carbon capture

process. The carbon capture terms mean a fraction of the steam (originally used for power generation) is drawn-off from the IP/LP crossover of a steam turbine [89] for MEA regeneration of the corresponding carbon capture plant. $\psi \in [0, 1]$ is the power penalty coefficient due to carbon capture. $\dot{m}_{\mathrm{B}}$ is the base mass flow rate of emitted CO_2 when a power plant operates under its nominal power output. $\mathcal{Q}_{\mathrm{reb}}$ is the specific reboiler duty for each unit of CO_2 emission. The capture level y is the fraction of CO_2 in the flue gas of a fossil-fuel power plant captured by a PCC plant. Note that some literature have discussed the optimal operation of an individual PCC plant and demonstrated that, to reduce the reboiler duty of carbon capture, the main degrees of freedom are the lean CO_2 loading and the carbon capture [17, 51]. Lean CO_2 loading is some variable which can be individually determined by the PCC plant while the capture level should be determined considering the fossil-fuel power plant operation. The profit model taking into account the power and carbon capture plant operation under electricity and CO_2 allowance market conditions is formulated as

$$\begin{aligned} r_{i,t+1} \triangleq & R_{\mathrm{E}i,t} - C_{i,t} \\ = & \lambda_{\mathrm{E}t} P_{\mathrm{g}i,t} T_{\mathrm{a}} - (\mathcal{N}_i + \frac{\mathcal{V}_{\mathrm{cc}i} \mathcal{E}_i y_{i,t} + f}{\eta_i}) P_{\mathrm{gg}i,t} T_{\mathrm{a}} \\ & - p_{\mathrm{s}t} q_{\mathrm{s}i,t} \end{aligned} \tag{5.51}$$

where the revenue $R_{\mathrm{E}i,t}$ comes from the electricity supply with the electricity price λ_{E} and the power generation P_{g}. Note that some variables, i.e., the market electricity price $\lambda_{\mathrm{E}t}$, the average power generation for one auction period $P_{\mathrm{g}i,t}$, the settled CO_2 allowance price $p_{\mathrm{s}t}$, and the winning bid quantity $q_{\mathrm{s}i,t}$, are known until the market auction window (i.e., auction t) is closed. Thus, the reward is written as $r_{i,t+1}$ rather than $r_{i,t}$.

5.3.2 Model of CO_2 allowance auctions

Apart from the constraints due to the operation of power plants with carbon capture (Eqs. (5.49) and (5.50)), other constraints origin from the auction mechanism of CO_2 allowances and are listed as follows (Eqs. (5.52)). Note that auctions are held fortnightly (T_{a}), which is identical to those auctions reported by Intercontinental Exchange (ICE) [19]. The detailed auction mechanism (Eq. (5.52)) is as follows. During

a two-hour bidding window of auction t, each auction participant should submit sealed bids to the auction platform without seeing the bids of other bidders. After the bidding window is closed to the participants, the platform will check whether any bidder behaves disorderly (Eqs. (5.55), (5.56) and (5.57)) and allocate the allowances in terms of a unified price settled according to bid prices. The bidders with higher bid prices will have higher priorities of procuring allowances. The settlement price is determined as the bid price of the marginal bidder(s) at which the sum of the bid quantities of those successful bidders equals or exceeds the total volume of allowances in that auction. Afterwards, all the successful bidders will be allocated with allowances at the unified settlement price. The participants with tie bids (i.e., identical bid prices) will be allocated CO_2 allowances with random priorities. This auction mechanism is represented by

$$\mathbf{y}_t = \mathcal{M}(p_{\mathrm{b}i,t}, q_{\mathrm{b}i,t}, p_{\mathrm{b}-i,t}, q_{\mathrm{b}-i,t}, y_{i,t}, y_{-i,t}, h_{i,t}, h_{-i,t}, \bar{v}_{\mathrm{a},t}, p_{\mathrm{r}t}) \tag{5.52}$$

where $\mathcal{M} : \mathcal{B} \times \mathcal{C} \times \mathcal{K} \to \mathcal{Y}$. $(p_{\mathrm{b}i,t}, p_{\mathrm{b}-i,t}, q_{\mathrm{b}i,t}, q_{\mathrm{b}-i,t}) \in \mathcal{B}$ is the submitted bids for all the plants, i.e., bidders, $(y_{i,t}, y_{-i,t}) \in \mathcal{C}$ is the carbon capture level of all the bidders, and $(\bar{v}_{\mathrm{a},t}, p_{\mathrm{r}t}) \in \mathcal{K}$ is some common knowledge which is a pair of the total CO_2 allowance volume to be auctioned, $\bar{v}_{\mathrm{a}t}$, and the price floor for CO_2 allowances, $p_{\mathrm{r}t}$ at auction t. Given these information, the auction platform will determine the auction settlement price, $p_{\mathrm{s}t}$ and the allowances offered to those successful bidders, $q_{\mathrm{s}i,t}$ as above, which is the auction result, $\mathbf{y}_t \triangleq (p_{\mathrm{s}t}, q_{\mathrm{s}0,t}, \cdots, q_{\mathrm{s}n-1,t}, \mathcal{I}_t)^T \in \mathcal{Y}$. $(p_{\mathrm{s}t}, q_{\mathrm{s}i,t})^T$ is termed as the allocation of CO_2 allowances for agent i. Note that when the bid price of any marginal plant is less than the reserve price $p_{\mathrm{r}t}$, the auction will be cancelled. And the allowance volume of this auction will be evenly distributed to subsequent $\tilde{n}$ auctions (Eqs. (5.53) and (5.54)). $\mathcal{I}_t \in \{0, 1\}$ is the indicator telling whether the allowances at auction t are allocated to bidders or distributed to subsequent auctions. The distribution method of CO_2 allowances is given by

$$\tilde{v}_{\mathrm{a}k+1} \leftarrow \tilde{v}_{\mathrm{a}k+1} + \mathcal{I}_t \bar{v}_{\mathrm{a},t}/\tilde{n}, \text{ for } k \in \{t, t+1, \cdots, t+\tilde{n}-1\} \tag{5.53}$$

$$\bar{v}_{\mathrm{a}t+1} = \tilde{v}_{\mathrm{a}t+1} + v_{\mathrm{a}t+1} \tag{5.54}$$

where $v_{\mathrm{a}t+1}$ is the scheduled allowance volume to be auctioned at time $t+1$ and $\tilde{v}_{\mathrm{a}t+1}$ is the cumulative allowance volume due to historical auction cancellations till auction $t+1$.

The auction platform should check the rationality of the behaviours of those bidders. In the event of monopolies, the platform will reject a particular bid for a participant when the sum of its bid quantity $q_{\mathrm{b}i,t}$ and the allowances in its holding account $h_{i,t}$ exceeds some holding limit $\bar{h}_i$, i.e.,

$$\mathcal{I}(h_{i,t} + q_{\mathrm{b}i,t} \leq \bar{h}_i) = \begin{cases} 1, \text{ if } h_{i,t} + q_{\mathrm{b}i,t} \leq \bar{h}_i \\ 0, \text{ otherwise} \end{cases} \tag{5.55}$$

$$q_{\mathrm{b}i,t} \leftarrow q_{\mathrm{b}i,t}\mathcal{I}(h_{i,t} + q_{\mathrm{b}i,t} \leq \bar{h}_i) \tag{5.56}$$

$$q_{\mathrm{s}i,t} \leftarrow q_{\mathrm{s}i,t}\mathcal{I}(h_{i,t} + q_{\mathrm{b}i,t} \leq \bar{h}_i). \tag{5.57}$$

In addition, adequate CO_2 allowances should be surrendered from the holding account of a bidder (i.e., the fossil-fuel power plant with carbon capture in this thesis) to cover CO_2 emission (Eqs. (5.58), (5.59), (5.60) and (5.61)) represented by

$$m_{i,t} = (P_{\mathrm{gg}i,t}/\eta_i)\mathcal{E}(1 - y_{i,t})T_{\mathrm{a}} \tag{5.58}$$

$$h_{i,t+1} = h_{i,t} + q_{\mathrm{s}i,t} - m_{i,t} \tag{5.59}$$

$$\mathcal{I}(h_{i,t+1} < 0) = \begin{cases} 1, \text{ if } h_{i,t+1} < 0 \\ 0, \text{ otherwise} \end{cases} \tag{5.60}$$

$$r_{i,t+1} \leftarrow r_{i,t+1} + p_{\mathrm{a}}h_{i,t+1}\mathcal{I}(h_{i,t+1} < 0). \tag{5.61}$$

During a specific auction t, for agent i, the CO_2 emission is denoted by $m_{i,t}$ and the winning CO_2 allowances is denoted by $q_{\mathrm{s}i,t}$. The CO_2 allowances in the holding account $h_{i,t}$ together with the winning CO_2 allowances should be able to cover the CO_2 emission in that auction. When inadequate allowances are surrendered (i.e., $h_{i,t+1} < 0$), the reward calculated with Eq. (5.51) suffers an extra penalty term $p_{\mathrm{a}}h_{i,t+1}$ from Eq.(5.61). p_{a} is the penalty due to non-compliance of allowance surrendering (€/allowance) for excess CO_2 emission.

5.3.3 Objective formulation

The objective function is formulated for each agent $i \in \{0, 1, \cdots, n-1\} \triangleq \mathbb{I}$ with n being the number of agents under the reinforcement learning framework. The action

of agent i is defined as

$$a_{i,t} \triangleq (q_{\mathrm{b}i,t}, p_{\mathrm{b}i,t}, y_{i,t}) \tag{5.62}$$

while the joint action of opponents $-i$ is defined as

$$a_{-i,t} \triangleq (q_{\mathrm{b}-i,t}, p_{\mathrm{b}-i,t}, y_{-i,t}). \tag{5.63}$$

A simple state of agent i is defined as

$$s_{i,t} \triangleq (P_{\mathrm{g}i,t-1}, h_{i,t}, v_{\mathrm{a}t}, \lambda_{\mathrm{E}t-1}, p_{\mathrm{s}t-1}, p_{\mathrm{r}t}, \kappa_{i,t}) \in \mathcal{S}_i \tag{5.64}$$

where $\kappa_{i,t}$ is the relative time index for the finite-time horizon M_i of agent i. For brevity, we use κ to denote the value of this relative time index for any agent at a certain time step, which should suffice

$$\kappa \leftarrow \begin{cases} \kappa + 1, \text{ if } \kappa \in \{1, 2, \cdots, M_i\} \\ 1, \text{ if otherwise} \end{cases}. \tag{5.65}$$

Based on the above state $s_{i,t}$ and $a_{i,t}$ for all $i \in \mathbb{I}$ at auction t, the transition to the next state $s_{i,t+1}$ can be fully derived as follows: According to Assumptions (i) and (ii), the transitions of $P_{\mathrm{g}i,t-1}$, $\lambda_{\mathrm{E}t-1}$, $v_{\mathrm{a}t}$ and $p_{\mathrm{r}t}$ to $P_{\mathrm{g}i,t}$, $\lambda_{\mathrm{E}t}$, $v_{\mathrm{a}t+1}$ and $p_{\mathrm{r}t+1}$ can be fully determined (Figures 5.1, 5.2, 5.3, and 5.4). The next-step $p_{\mathrm{s}t}$ can be determined by Eq. (5.52) with the inputs, $h_{i,t}$, $h_{-i,t}$, $v_{\mathrm{a}t}$, $p_{\mathrm{r}t}$, $a_{i,t}$, and $a_{-i,t}$. The next-step holding account allowances $h_{i,t+1}$ can be derived based on Eq. (5.59). Thus, all the state entries of $s_{i,t+1}$ can be obtained from $s_{i,t}$ and $a_{i,t}$ for all $i \in \mathbb{I}$. Note that the state $s_{i,t}$ is constructed with a mixture of discrete and continuous state entries (Eq. (5.64)). To implement a tabular Sarsa or Sarsa(λ) learning algorithm (Table 5.1 or 5.2), state aggregation can be used, which partitions the state space $\mathcal{S}_i$ into a finite set of n_i subsets, i.e.,

$$\mathbf{partition}(\mathcal{S}_i) \triangleq \{\mathcal{S}_{i,1}, \mathcal{S}_{i,2}, \cdots, \mathcal{S}_{i,n_i}\}. \tag{5.66}$$

By defining $\bar{\mathcal{S}}_i \triangleq \{1, 2, \cdots, n_i\}$, there is a feature function $\mathcal{F}_{\mathrm{s}i} : \mathcal{S}_i \to \bar{\mathcal{S}}_i$

$$\mathcal{F}_{\mathrm{s}i}(s_{i,t}) = \bar{s}, \text{ if } s_{i,t} \in \mathcal{S}_{i,\bar{s}} \tag{5.67}$$

where index $\bar{s}$ is the discrete state index whereas $\bar{\mathcal{S}}_i$ is the finite state index set for agent i. Note that we only consider some finite discrete action set $\mathcal{A}_i$, which means the feature function w.r.t. any action can be represented by $\mathcal{F}_{\mathrm{a}i} : \mathcal{A}_i \to \bar{\mathcal{A}}_i$

$$\mathcal{F}_{\mathrm{a}i}(a_{i,t}) = \bar{a}, \text{ if } a_{i,t} \in \mathcal{A}_{i,\bar{a}} \tag{5.68}$$

with $\bar{a} \in \{1, 2, \cdots, |\mathcal{A}_i|\} \triangleq \bar{\mathcal{A}}_i$. Thus, the estimate of any action value function $q_{\pi_i,i}(s, a)$ with $(s, a) \in \mathcal{S}_i \times \mathcal{A}_i$ can be approximated by

$$q_{i,\pi_i}(s, a) \approx Q_i(\mathcal{F}_{\mathrm{s}i}(s), \mathcal{F}_{\mathrm{a}i}(a)) = Q_i(\bar{s}, \bar{a}). \tag{5.69}$$

In comparison with Eq.(5.14), the action-value function of agent i is defined as

$$q_{\pi_i,i}(s, a) \triangleq \mathbb{E}_\pi[\sum_{k=t}^{\infty} \gamma^{k-t} R_{i,k+1} | S_{i,t} = s, A_{i,t} = a] \tag{5.70}$$

where $s \in \mathcal{S}_i$, $a \in \mathcal{A}_i$, and $i \in \mathbb{I}$. We will consider the expected return of any integrated carbon capture plant i over a finite-time horizon, which indicates

$$q_{\pi_i,i}(s, a) = \mathbb{E}_\pi[\sum_{k=t}^{T_{i,\kappa}-1} \gamma^{k-t} R_{i,k+1} | S_{i,t} = s, A_{i,t} = a] \tag{5.71}$$

where $T_{i,\kappa} = t + M_i - \kappa$ is the terminal time index, M_i is the non-terminal time horizon of agent i, and $\kappa \in \{1, 2, \cdots, M_i\}$. The terminal state set $\mathcal{S}_i \cap \mathcal{S}_i^+$ is any state s (Eq. (5.64)) whose entry $\kappa \notin \{1, 2, \cdots, M_i\}$. In this thesis, all the agents has a unified time horizon, which infers $M_i = M_{i+1}$ for $\forall i$. The modified Sarsa learning algorithm with state aggregation is presented in Table 5.5. Similar modifications can be applied to the online Sarsa(λ) algorithm and is presented in Table 5.6.

5.4 Simulation results

5.4.1 Simulation settings

In the following simulation, simple CO_2 allowance auctions are held among five carbon capture plants integrated with fossil-fuel power plants. The power plants should fulfil the power generation tasks with a higher priority than carbon capture (Assumption (iv)) but subject to some emission penalties (Eqs. (5.58), (5.59), (5.60) and (5.61)).

Table 5.5: Sarsa TD control with ε-greedy policy and state aggregation.

Schedule step size α_i and the non-greedy probability ε_i of all $i \in \mathbb{I}$
Initialise $Q_i(s,a)$ for $(s,a) \in \mathcal{S}_i \times \mathcal{A}_i$ arbitrarily of all $i \in \mathbb{I}$
Initialise $Q_i(s,\cdot) = 0$ for $s \in \mathcal{S}_i \cap \mathcal{S}_i^+$ of all $i \in \mathbb{I}$
while true
  Initialise S_i of all $i \in \mathbb{I}$
  Select A_i based on $\pi_i(\cdot|S_i)$ (where π_i is an ε-greedy policy) of all $i \in \mathbb{I}$
  do
    Take action A_i, observe new state S_i', and obtain reward R_i of all $i \in \mathbb{I}$
    Select A_i' based on $\pi_i(\cdot|S_i')$ (where π_i is an ε-greedy policy) of all $i \in \mathbb{I}$
    $Q_i(\mathcal{F}_{\mathrm{s}i}(S_i), \mathcal{F}_{\mathrm{a}i}(A_i)) \leftarrow Q_i(\mathcal{F}_{\mathrm{s}i}(S_i), \mathcal{F}_{\mathrm{a}i}(A_i))$
      $+\alpha_i(R_i + \gamma Q_i(\mathcal{F}_{\mathrm{s}i}(S_i'), \mathcal{F}_{\mathrm{a}i}(A_i')) - Q(\mathcal{F}_{\mathrm{s}i}(S_i), \mathcal{F}_{\mathrm{s}i}(A_i)))$
    $S_i \leftarrow S_i'$, $A_i \leftarrow A_i'$ of all $i \in \mathbb{I}$
  until S_i is a terminal state

Table 5.6: Sarsa(λ) control with ε-greedy policy and state aggregation.

Schedule step size α_i and the non-greedy probability ε_i of all $i \in \mathbb{I}$
Initialise $Q_i(s,a)$ for $(s,a) \in \mathcal{S}_i \times \mathcal{A}_i$ arbitrarily of all $i \in \mathbb{I}$
Initialise $Q_i(s,\cdot) = 0$ for $s \in \mathcal{S}_i \cap \mathcal{S}_i^+$ of all $i \in \mathbb{I}$
while true
  Initialise S_i of all $i \in \mathbb{I}$
  Select A_i based on $\pi_i(\cdot|S_i)$ (where π_i is an ε-greedy policy) of all $i \in \mathbb{I}$
  do
    Take action A_i, observe new state S_i', and obtain reward R_i of all $i \in \mathbb{I}$
    Select A_i based on $\pi_i(\cdot|S_i)$ (where π_i is an ε-greedy policy) of all $i \in \mathbb{I}$
    $\delta_i \leftarrow R_i + \gamma Q_i(\mathcal{F}_{\mathrm{s}i}(S_i'), \mathcal{F}_{\mathrm{a}i}(A_i')) - Q_i(\mathcal{F}_{\mathrm{s}i}(S_i), \mathcal{F}_{\mathrm{a}i}(A_i))$ of all $i \in \mathbb{I}$
    $z_i(\mathcal{F}_{\mathrm{s}i}(S), \mathcal{F}_{\mathrm{a}i}(A)) \leftarrow z_i(\mathcal{F}_{\mathrm{s}i}(S), \mathcal{F}_{\mathrm{a}i}(A)) + 1$ of all $i \in \mathbb{I}$
    for each (s,a) in $\mathcal{S}_i \times \mathcal{A}_i$
      $Q_i(\mathcal{F}_{\mathrm{s}i}(s), \mathcal{F}_{\mathrm{a}i}(a)) \leftarrow Q_i(\mathcal{F}_{\mathrm{s}i}(s), \mathcal{F}_{\mathrm{a}i}(a)) + \alpha_i \delta_i z_i(\mathcal{F}_{\mathrm{s}i}(s), \mathcal{F}_{\mathrm{a}i}(a))$
      $z_i(\mathcal{F}_{\mathrm{s}i}(s), \mathcal{F}_{\mathrm{a}i}(a)) \leftarrow \gamma\lambda z_i(\mathcal{F}_{\mathrm{s}i}(s), \mathcal{F}_{\mathrm{a}i}(a))$
    $S_i \leftarrow S_i'$, $A_i \leftarrow A_i'$ of all $i \in \mathbb{I}$
  until S_i is a terminal state

For convenience, these plants are indexed by $i \in \mathbb{I} = \{0,1,2,3,4,5\}$ and use either coal-fired or natural-gas technology (Tables 5.3 and 5.4) for power generation. The power-generation technologies together with the power capacities for each plant are summarised in Table 5.7. Given the power generation, electricity price, scheduled reserve price and scheduled allowance profiles (Figure 5.1, 5.2, 5.3 and 5.4), any MEA-based PCC plant integrated with a fossil-fuel power plant (or plant i for short) should determine its carbon capture level and the bid submitted to the CO_2 allowance auction

during each auction period, which is 14 days, i.e. $14 \times 24 = 336$ hours. It should be noted that the power generation in Figure 5.2 is the average power generation over any auction period, from which we can derive the associated electricity production (MWh).

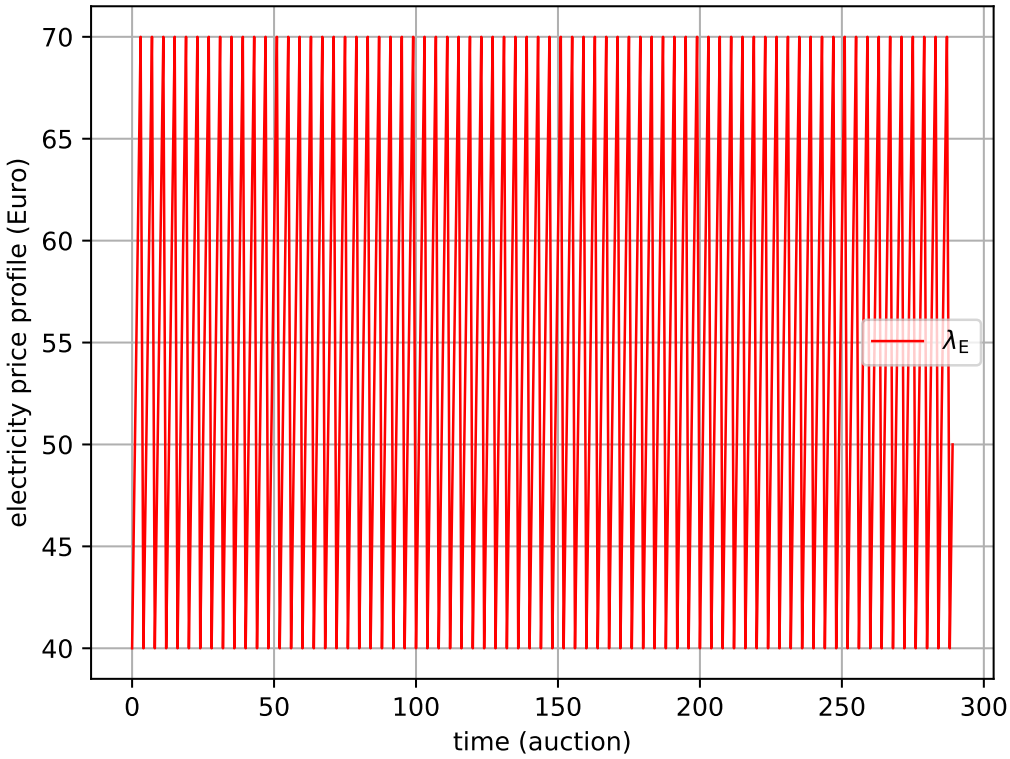


Figure 5.1: The electricity price profile.

To mitigate the curse of dimensions, state aggregation is used which partitions the state space $\mathcal{S}_i$ (Eq. (5.64)) into a finite set of subsets (Eq. (5.66)). The state aggregation for the preceding decision making problem is achieved as follows. Based on the power generation profiles of each plant (Table (5.8), partitions are applied for each state entry in Eq. (5.64). For instance, although the power generation can be any value in $[0, 650.0]$ for plant i with power capacity 650 MW, the agent with Sarsa or Sarsa(λ) learning algorithm will only refer to the power output as one of three different domains, i.e., domain 1 (less than 300.3 MW), domain 2 (between 300.3 and 360.1 MW) and domain 3 (greater than 360.1 MW). These domains for power generations are fixed based the average power generations and the maximum power output for each power plant. Generally, the domains are identified based on the possible values which can be

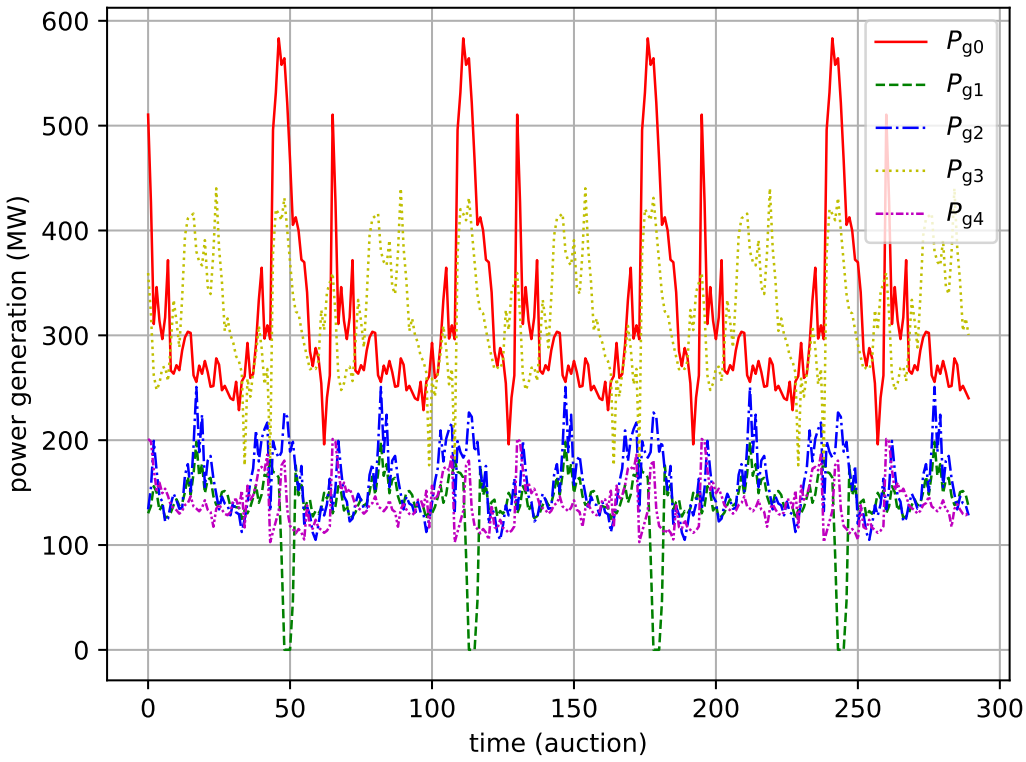


Figure 5.2: The power generation profile.

taken by each state entry in Eq. (5.64). Further details of the partitions are shown in Table 5.8 which implies

$$\begin{aligned}
\textbf{partion}(\mathcal{S}_{P_{\mathrm{g},i}}) &= \{\mathcal{S}^1_{P_{\mathrm{g},i}}, \mathcal{S}^2_{P_{\mathrm{g},i}}, \mathcal{S}^3_{P_{\mathrm{g},i}}\} \\
\textbf{partion}(\mathcal{S}_{h_i}) &= \{\mathcal{S}^1_{h_i}, \mathcal{S}^2_{h_i}\} \\
\textbf{partion}(\mathcal{S}_{v_\mathrm{a}}) &= \{\mathcal{S}^1_{v_\mathrm{a}}, \mathcal{S}^2_{v_\mathrm{a}}, \mathcal{S}^3_{v_\mathrm{a}}, \mathcal{S}^4_{v_\mathrm{a}}, \mathcal{S}^5_{v_\mathrm{a}}\} \\
\textbf{partion}(\mathcal{S}_{\lambda_\mathrm{E}}) &= \{\mathcal{S}^1_{\lambda_\mathrm{E}}, \mathcal{S}^2_{\lambda_\mathrm{E}}, \mathcal{S}^3_{\lambda_\mathrm{E}}, \mathcal{S}^4_{\lambda_\mathrm{E}}\} \\
\textbf{partion}(\mathcal{S}_{p_\mathrm{s}}) &= \{\mathcal{S}^1_{p_\mathrm{s}}, \mathcal{S}^2_{p_\mathrm{s}}, \mathcal{S}^3_{p_\mathrm{s}}\} \\
\textbf{partion}(\mathcal{S}_{p_\mathrm{r}}) &= \{\mathcal{S}^1_{p_\mathrm{r}}, \mathcal{S}^2_{p_\mathrm{r}}, \mathcal{S}^3_{p_\mathrm{r}}, \mathcal{S}^4_{p_\mathrm{r}}, \mathcal{S}^5_{p_\mathrm{r}}\} \\
\textbf{partion}(\mathcal{S}_{\kappa_i}) &= \{\mathcal{S}^1_{\kappa_i}\}
\end{aligned}$$

should be partitions of each state entry where $\mathcal{S}^1_{\kappa_i} = \{1\}$. Subsequently, we can define a one-to-one mapping which points $(j_1, j_2, \cdots, j_7)$ to $\bar{s}$ for $\forall j_1, j_2, \cdots, j_7$. Thus, the states

$$s_{i,t} \in \mathcal{S}_{i,\bar{s}} \triangleq \mathcal{S}^{j_1}_{P_{\mathrm{g},i}} \times \mathcal{S}^{j_2}_{h_i} \times \mathcal{S}^{j_3}_{v_\mathrm{a}} \times \mathcal{S}^{j_4}_{\lambda_\mathrm{E}} \times \mathcal{S}^{j_5}_{p_\mathrm{s}} \times \mathcal{S}^{j_6}_{p_\mathrm{r}} \times \mathcal{S}^{j_7}_{\kappa_i}$$

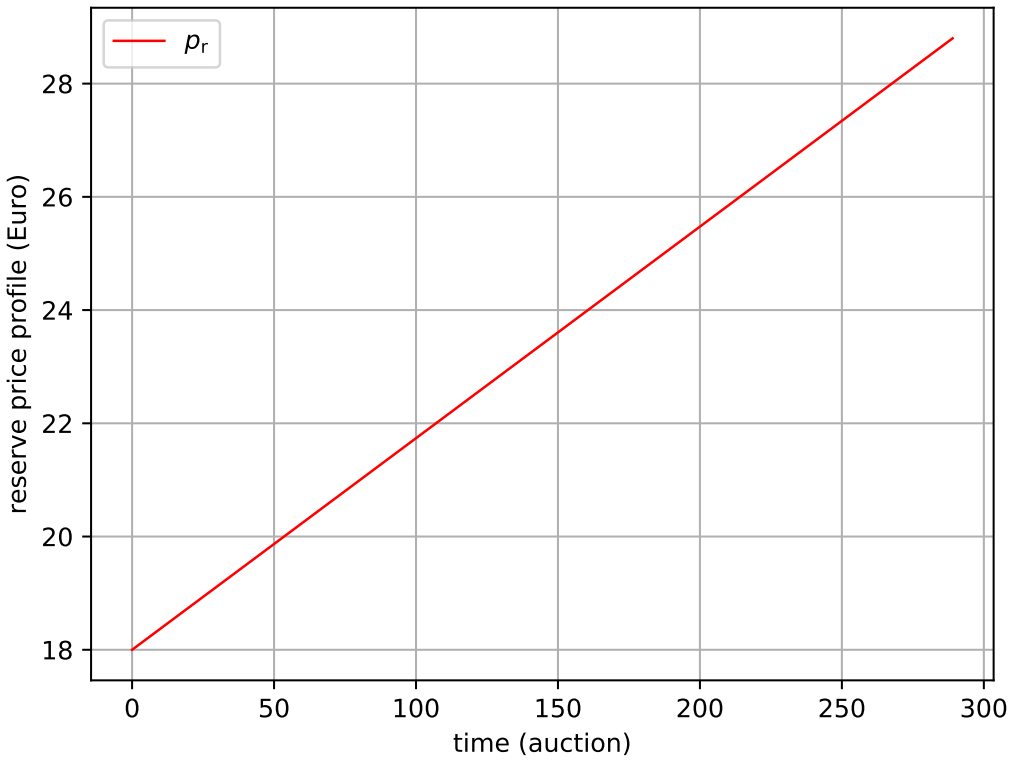


Figure 5.3: CO_2 allowance reserve price profile.

are grouped as one single state indexed by $\bar{s}$ through Eq. (5.67) where $\mathcal{S}_{i,\bar{s}}$ is any subset defined based on **partition**$(\mathcal{S}_i)$ (Eq. (5.66)). Then, the Sarsa or Sarsa(λ) algorithm can be implemented based on the features $F_{si}(s_{i,t}) \in \bar{\mathcal{S}}_i$ (Tables 5.5 and 5.6) rather than the original states $s_{i,t} \in \mathcal{S}_i$ (Tables 5.1 and 5.2). Additionally, the finite set for the discrete action $\mathcal{A}_i$ is given by Table 5.9. In Table 5.10, for $\forall i \in \mathbb{I}$, α_i is changed from α_{start} to α_{end}; ε_i is varied from $\varepsilon_{\text{start}}$ to ε_{end}. Both α_i and ε_i are changed every time the state transition happens, which implies we only consider the unified learning step and the non-greedy probability for all the state-action pair and the state, respectively. $\alpha_i^{N_s}$ and $\varepsilon_i^{N_s}$ are geometric sequences $\alpha_{i,0}, \alpha_{i,1}, \cdots, \alpha_{i,N_s-1}$ and $\varepsilon_{i,0}, \varepsilon_{i,1}, \cdots, \varepsilon_{i,N_s-1}$, respectively, which implies

$$\alpha_{i,t+1} = \alpha_{i,t} \exp\left(\frac{1}{N_s - 1} \log \frac{\alpha_{\text{end}}}{\alpha_{\text{start}}}\right) \tag{5.72}$$

$$\varepsilon_{i,t+1} = \varepsilon_{i,t} \exp\left(\frac{1}{N_s - 1} \log \frac{\varepsilon_{\text{end}}}{\varepsilon_{\text{start}}}\right) \tag{5.73}$$

with $\alpha_{i,0} = \alpha_{\text{start}}$ and $\varepsilon_{i,0} = \varepsilon_{\text{start}}$. N_s is the number of the learning steps of each agent. The discount rate is supposed to be $\gamma = 1/(1 + 8\%) = 0.926$. Besides these tuning parameters, the initialisation method when a terminal state reaches should be

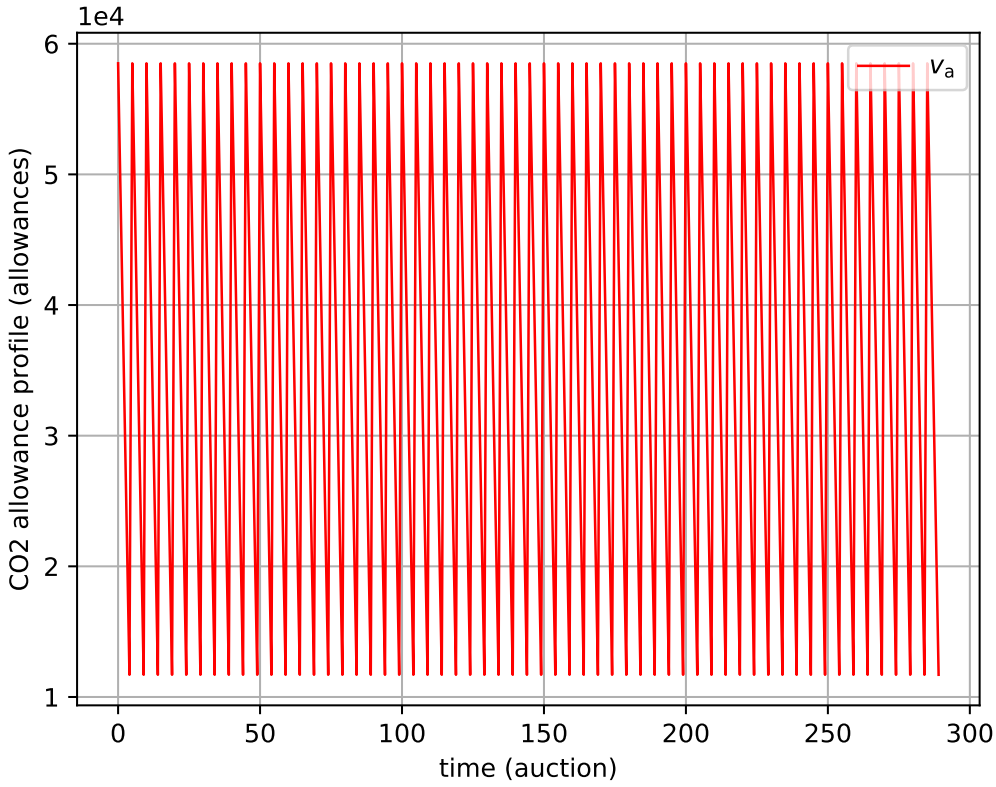


Figure 5.4: CO_2 allowance volume profile.

specified (Tables 5.5 and 5.6). To visit as many states as possible, random samples for each scheduled profile should be separately selected. For our case, during any initialisation, a random sample index is generated which points to a sample pair of the power generation and the electricity price profiles in Figure 5.1, and Figure 5.2. Similarly, two more independent sample indexes are generated to select the sample in the reserve price or allowance profiles in Figure 5.3 and Figure 5.4. In addition, m_t is initialised in terms of Eq. (5.65) while $h_{i,t}$ is initialised uniformly random from the per-unit domain $[0.0, 0.2]$ for all $i \in \mathbb{I}$ w.r.t. the bases in Table 5.8.

5.4.2 Training of Sarsa and Sarsa(λ)

For this case study, we consider the learning performances of both Sarsa and Sarsa(λ) algorithms based on the settings in the previous subsection. The original rewards observed by each plant during the training process are shown in Figure 5.5 and Figure 5.6. To demonstrate the performances for each plant $i \in \mathbb{I}$, the moving-average

reward of an agent for every 100 steps defined by

$$r_{\text{mavg}i,t} = \frac{1}{100} \sum_{k=t-99}^{t} r_{i,k} \tag{5.74}$$

and the average reward defined by

$$r_{\text{avg}i,t} = \frac{1}{t} \sum_{k=1}^{t} r_{i,k} \tag{5.75}$$

are plotted in Figure 5.7 and Figure 5.8, respectively, from which, it can be found that the rewards obtained through both learning algorithms gradually increase for each fossil-fuel power plant with carbon capture. As the learning moves forward, the agents using either Sarsa or Sarsa(λ) learning algorithm show some for-profit properties. The average reward differences between the Sarsa and Sarsa(λ) training (Figure 5.9) indicate that the Sarsa(λ) agents learn much more quickly during some initial stage, which is roughly the first 25000 learning steps. Afterwards, since the online Sarsa(λ) is just approximately equivalent to the corresponding offline algorithm, the performances of the Sarsa(λ)-based leaning agents become worse than the Sarsa-based learning agents gradually. In addition, although the continuous state space $\mathcal{S}_i$ is partitioned into discrete subsets, the total number of the non-terminal states of each agent in terms of Table 5.8 is $3 \times 2 \times 5 \times 4 \times 3 \times 5 \times 1 = 1800$. Currently, the bidding problem is solved centrally by one computer which means we should consider the decision-making problems of all the five power plants with carbon capture (Table 5.7) simultaneously. This leads to some high computation burden: The computation time of the Sarsa learning algorithm is about 379 seconds while that of the Sarsa(λ) learning algorithm is about 8347 seconds. Note that for one learning step, the Sarsa(λ) algorithm updates state values for each state-action pair $(s, a) \in \mathcal{S}_i \times \mathcal{A}_i$ of all $i \in \mathbb{I}$ while the Sarsa algorithm only updates the visited state-action pair. To reduce the computation burden, during each state transition, we only update the state-action pairs whose eligibility traces $z_t(s, a)$ is greater than a predefined tolerance which is 10^{-4}.

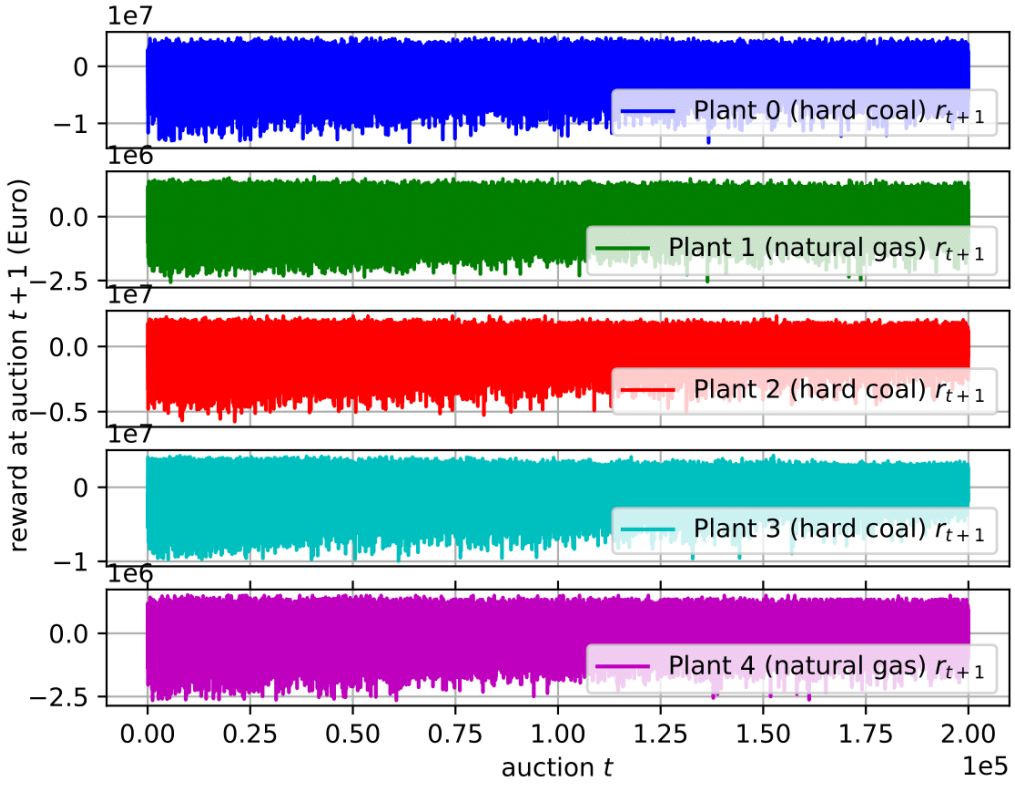


Figure 5.5: Rewards of Sarsa during training.

Table 5.7: Power plant settings of auction participants.

Plant ID	Technology	Power capacity
Plant 0	Hard coal	650 MW
Plant 1	Natural gas	650 MW
Plant 2	Hard coal	325 MW
Plant 3	Hard coal	650 MW
Plant 4	Natural gas	650 MW

5.4.3 Testing of Sarsa and Sarsa(λ)

In this subsection, the test results are demonstrated based on the policies obtained with Sarsa and Sarsa(λ) in the previous subsection. The electricity price λ_{E}, CO_2 allowance reserve price p_{r} and scheduled allowance volume v_{a} are kept to be 70 €/MWh, 18 €/allowance and 23399 allowances, respectively. The entries of those actions taken by each Sarsa or Sarsa(λ) agent are shown in Figure 5.10, Figure 5.11 and Figure 5.12 while the observed reward sequences are plotted in Figure 5.13. To compare the performances of different learning agents, average reward sequences are plotted as before

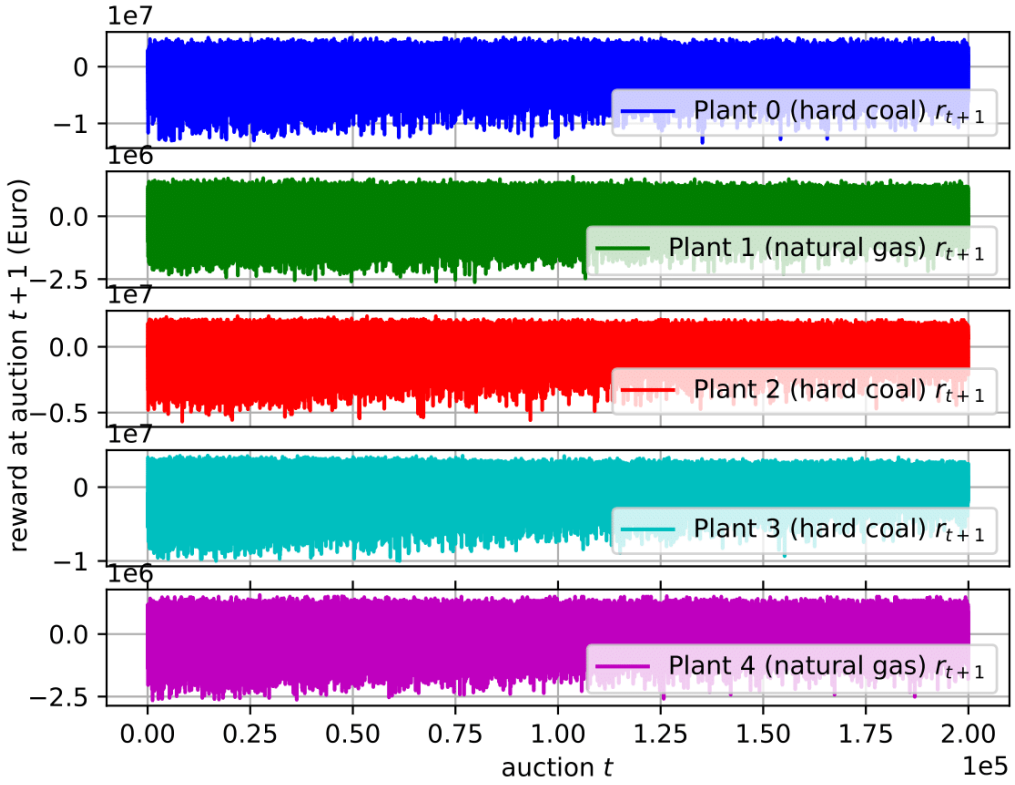


Figure 5.6: Rewards of Sarsa(λ) during training.

(Figure 5.14). As one can see from the figures, the performances of the Sarsa leaning algorithm during testing are quite similar to the performances of the Sarsa(λ) learning algorithm. However, since the Sarsa learning algorithm can be implemented with less computation time (i.e., 379 seconds as mentioned before), it is more favourable.

5.4.4 Effects of the non-terminal time horizon on Sarsa and Sarsa(λ)

In the previous cases, we assume that $M_i = M = 1$ for all $i \in \mathbb{I}$ as in Table 5.10. In this subsection, we consider $M_i = 3$ but keep the other tuning parameters unchanged. Similar to those previous cases, the moving-average rewards and the average reward differences are shown in Figure 5.15 and Figure 5.16, respectively. Note that when $M_i > 1$, some actions may be profitable but only in the first few steps in an episode. The reason is that an agent or the plant may resort to its holding account CO_2 allowances to cover its CO_2 emission. These kinds of actions get no instant penalties.

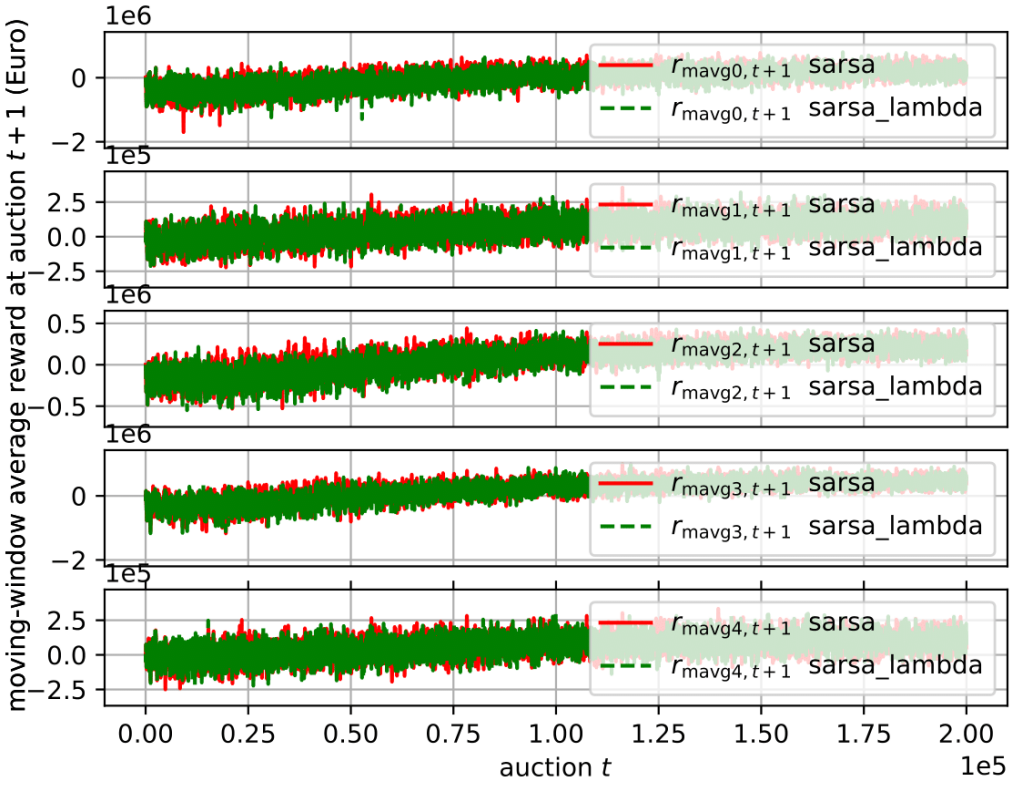


Figure 5.7: Moving-average rewards (every 100 steps) of Sarsa and Sarsa(λ) during training.

And as shown during the initial 50000 learning steps both Sarsa and Sarsa(λ) actually are misled by such kinds of profits. However, as learning steps move forward, large penalties will occur to an agent when the allowances in its holding account exhausted. Similarly to the EU ETS, non-compliance will cause an additional penalty p_{a} of 100 €/allowance for each ton of excess CO_2 emission. At this critical penalty point, Sarsa learning algorithm will only update the action value of the current visited state-action pair while Sarsa(λ) will update the action values of all the state-action pairs whose eligibility traces are greater than some predefined tolerance. From Figure 5.16, we can observe that Sarsa(λ) may initially obtain some lower rewards and deviated away from the rewards gained by Sarsa. However, when the learning steps move forward, Sarsa(λ) quickly find their appropriate actions, and finally, even find some policies with better performances and fewer steps than the Sarsa learning algorithm.

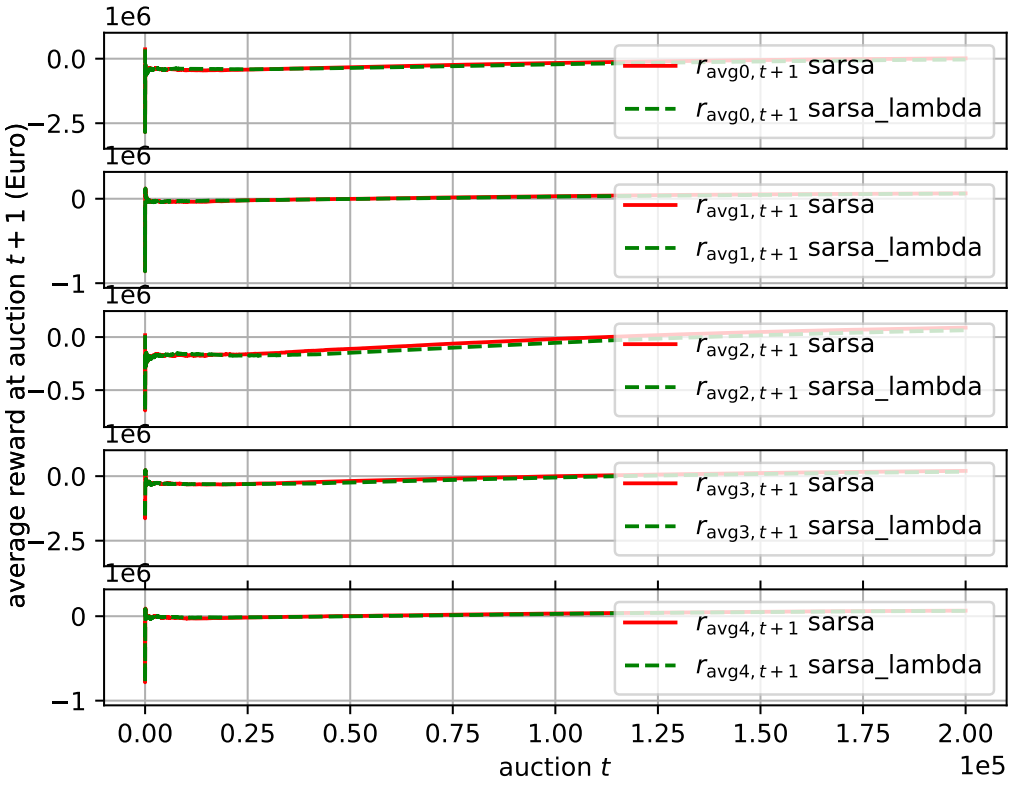


Figure 5.8: Average rewards of Sarsa and Sarsa(λ) during training.

5.5 Summary

In this chapter, we have formulated some for-profit agents on behalf of the MEA-based PCC plants which should determine their optimal carbon capture levels during each CO_2 allowance auction. This decision-making issue considers three sets of factors which include the time-varying power plant operation, the electricity market conditions and the CO_2 allowance market conditions. More specifically, the power generations (Figure 5.2), the electricity prices (Figure 5.1), allowance reserve prices (Figure 5.3), allowance volumes (Figure 5.4) and the auction mechanism (Eq. (5.52)) are considered to formulate a profit-maximum objective (Eq. (5.71)).

It is demonstrated that both the Sarsa and Sarsa(λ) learning algorithms can learn some for-profit policies. On the one hand, the Sarsa learning algorithm has a low computation burden and it performs better than Sarsa(λ) when there is no delayed penalties (Figure 5.9). On the other hand, Sarsa(λ) can converge fast with fewer learning steps than Sarsa and is able to deal with the delayed penalties (Figure 5.16). These

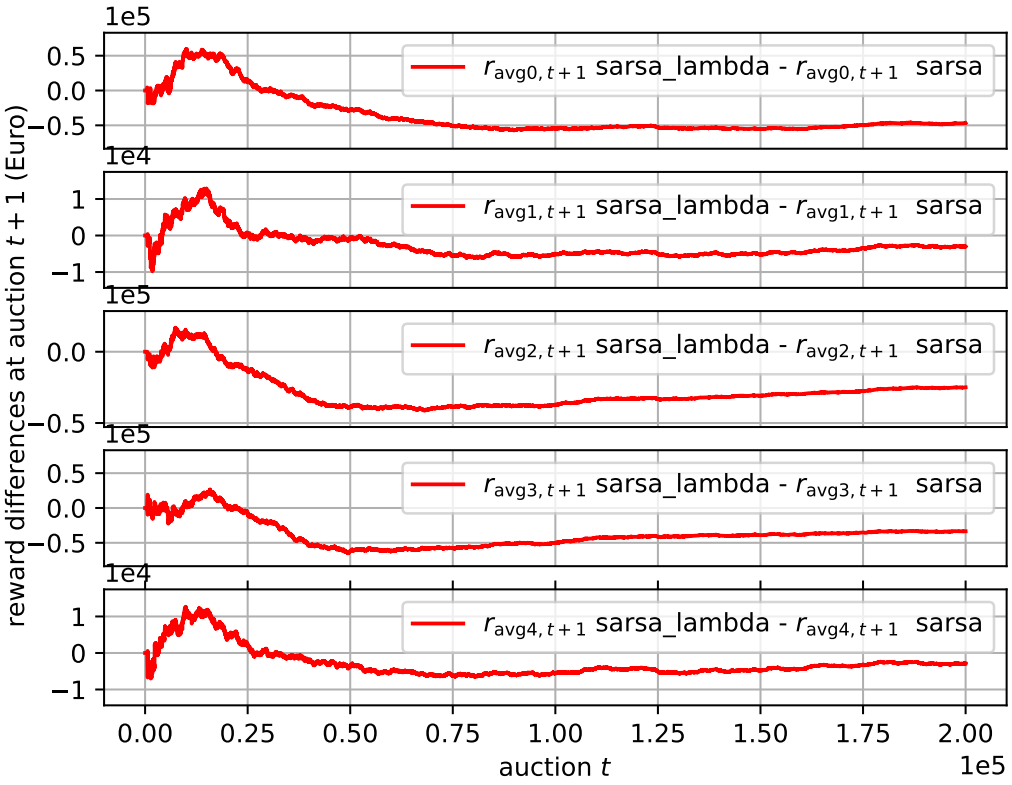


Figure 5.9: Average reward differences of Sarsa and Sarsa(λ) during training.

imply that if the data can be generated cheaply and there exist no delayed penalties, Sarsa learning algorithm should be preferred; otherwise, Sarsa(λ) is preferred. Furthermore, when we are facing some time-varying environments, Sarsa(λ) is usually prevalent due to its fast initialisation characteristics. However, it is also worthwhile noting that Sarsa(λ) is more computation-intensive than the Sarsa learning algorithm.

Table 5.8: State settings.

State	Base	Unit	Partitions (per unit)	Partitions
$P_{\mathrm{g}0,t-1}$	650	MW	$\{(-\infty, 0.462]$, $(0.462, 0.554]$, $(0.554, \infty)\}$	$\{(-\infty, 300.3]$, $(300.3, 360.1]$, $(360.1, \infty)\}$
$P_{\mathrm{g}1,t-1}$	650	MW	$\{(-\infty, 0.154]$, $(0.154, 0.262]$, $(0.262, \infty)\}$	$\{(-\infty, 100.1]$, $(100.1, 170.3]$, $(170.3, \infty)\}$
$P_{\mathrm{g}2,t-1}$	325	MW	$\{(-\infty, 0.462]$, $(0.462, 0.554]$, $(0.554, \infty)\}$	$\{(-\infty, 150.2]$, $(150.2, 180.1]$, $(180.1, \infty)\}$
$P_{\mathrm{g}3,t-1}$	650	MW	$\{(-\infty, 0.462]$, $(0.462, 0.540]$, $(0.540, \infty)\}$	$\{(-\infty, 300.3]$, $(300.3, 351.0]$, $(351.0, \infty)\}$
$P_{\mathrm{g}4,t-1}$	650	MW	$\{(-\infty, 0.192]$, $(0.192, 0.246]$, $(0.246, \infty)\}$	$\{(-\infty, 124.8]$, $(124.8, 159.9]$, $(159.9, \infty)\}$
$h_{0,t-1}$	395570	allowances	$\{(-\infty, 0.2]$, $(0.2, \infty)\}$	$\{(-\infty, 79114.0]$, $(79114.0, \infty)\}$
$h_{1,t-1}$	85155	allowances	$\{(-\infty, 0.2]$, $(0.2, \infty)\}$	$\{(-\infty, 17031.0]$, $(17031.0, \infty)\}$
$h_{2,t-1}$	198400	allowances	$\{(-\infty, 0.2]$, $(0.2, \infty)\}$	$\{(-\infty, 39680.0]$, $(39680.0, \infty)\}$
$h_{3,t-1}$	395570	allowances	$\{(-\infty, 0.2]$, $(0.2, \infty)\}$	$\{(-\infty, 79114.0]$, $(79114.0, \infty)\}$
$h_{4,t-1}$	85155	allowances	$\{(-\infty, 0.2]$, $(0.2, \infty)\}$	$\{(-\infty, 17031.0]$, $(17031.0, \infty)\}$
$v_{\mathrm{a}t}$	234000	allowances	$\{(-\infty, 0.075]$, $(0.075, 0.125]$, $(0.125, 0.175]$, $(0.175, 0.225]$, $(0.225, \infty)\}$	$\{(-\infty, 17550.0]$, $(17550.0, 29250.0]$, $(29250.0, 40950.0]$, $(40950.0, 52650.0]$, $(52650.0, \infty)\}$
$\lambda_{\mathrm{E}t-1}$	54.9	€/MWh	$\{(-\infty, 0.82]$, $(0.82, 1.0]$, $(1.0, 1.18]$, $(1.18, \infty)\}$	$\{(-\infty, 44.9]$, $(44.9, 54.9]$, $(54.9, 64.9]$, $(64.9, \infty)\}$
$p_{\mathrm{s}t-1}$	18.0	€/allowance	$\{(-\infty, 1.2]$, $(1.2, 1.6]$, $(1.6, \infty)\}$	$\{(-\infty, 21.6]$, $(21.6, 28.8]$, $(28.8, \infty)\}$
$p_{\mathrm{r}t}$	18.0	€/allowance	$\{(-\infty, 1.12]$, $(1.12, 1.24]$, $(1.24, 1.36]$, $(1.36, 1.48]$, $(1.48, \infty)\}$	$\{(-\infty, 20.2]$, $(20.2, 22.3]$, $(22.3, 24.5]$, $(24.5, 26.64]$, $(26.64, \infty)\}$

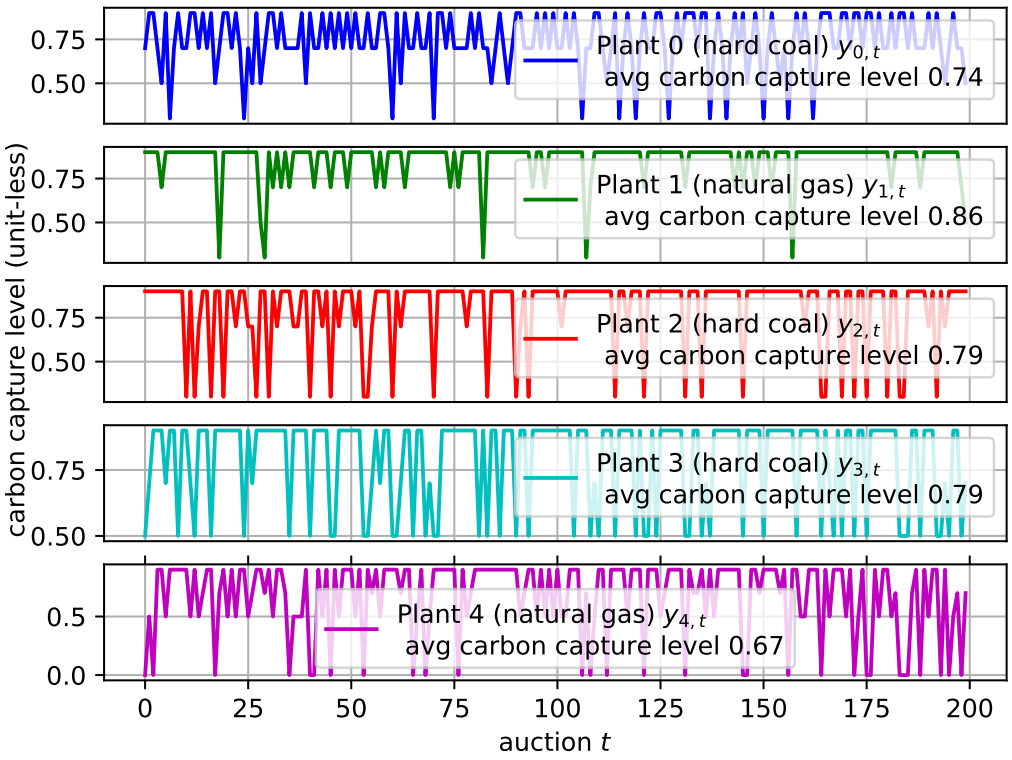
Table 5.9: Action settings.

Action	Base	Unit	Partitions (per unit)	Partitions
$q_{\mathrm{b}0,t}$	79114	allowances	{{0.1}, {0.5}, {0.9}}	{{8000}, {40000}, {71500}}
$q_{\mathrm{b}1,t}$	17031	allowances	{{0.1}, {0.5}, {0.9}}	{{2000}, {9000}, {15500}}
$q_{\mathrm{b}2,t}$	39680	allowances	{{0.1}, {0.5}, {0.9}}	{{4000}, {20000}, {36000}}
$q_{\mathrm{b}3,t}$	39680	allowances	{{0.1}, {0.5}, {0.9}}	{{8000}, {40000}, {71500}}
$q_{\mathrm{b}4,t}$	17031	allowances	{{0.1}, {0.5}, {0.9}}	{{2000}, {9000}, {15500}}
${p_{\mathrm{b}i,t}}^*$	18.0	€/allowance	{{1.0}, {1.4}, {1.8}}	{{18.0}, {25.2}, {32.4}}
${y_{i,t}}^*$	1.0	unit-less	{{0.0}, {0.3}, {0.5}, {0.7}, {0.9}}	{{0.0}, {0.3}, {0.5}, {0.7}, {0.9}}

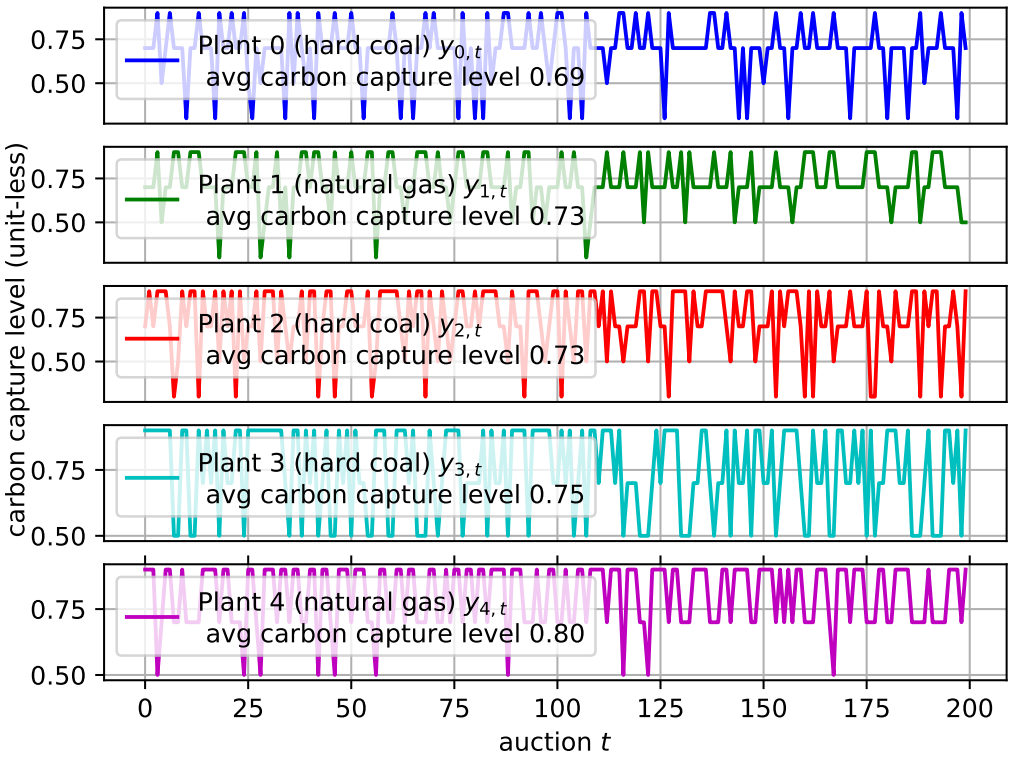
*where $i \in \mathbb{I}$.

Table 5.10: Base-case Sarsa and Sarsa(λ) tuning parameters of agents.

	Sarsa	Sarsa(λ)
α_{start}	0.9	0.9
α_{end}	0.1	0.1
$\varepsilon_{\mathrm{start}}$	1.0	1.0
$\varepsilon_{\mathrm{end}}$	0.01	0.01
λ	N/A	0.6
γ	0.926	0.926
M	1	1
N_{s}	200000	200000
tolerance	N/A	10^{-4}

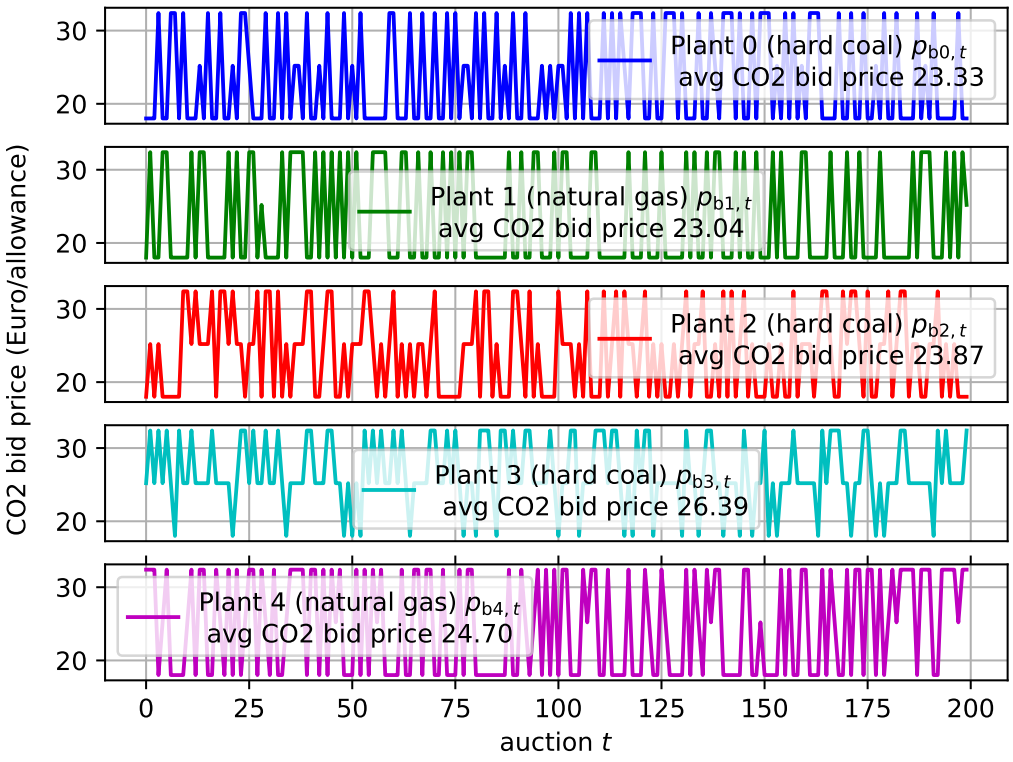


(a) Sarsa.

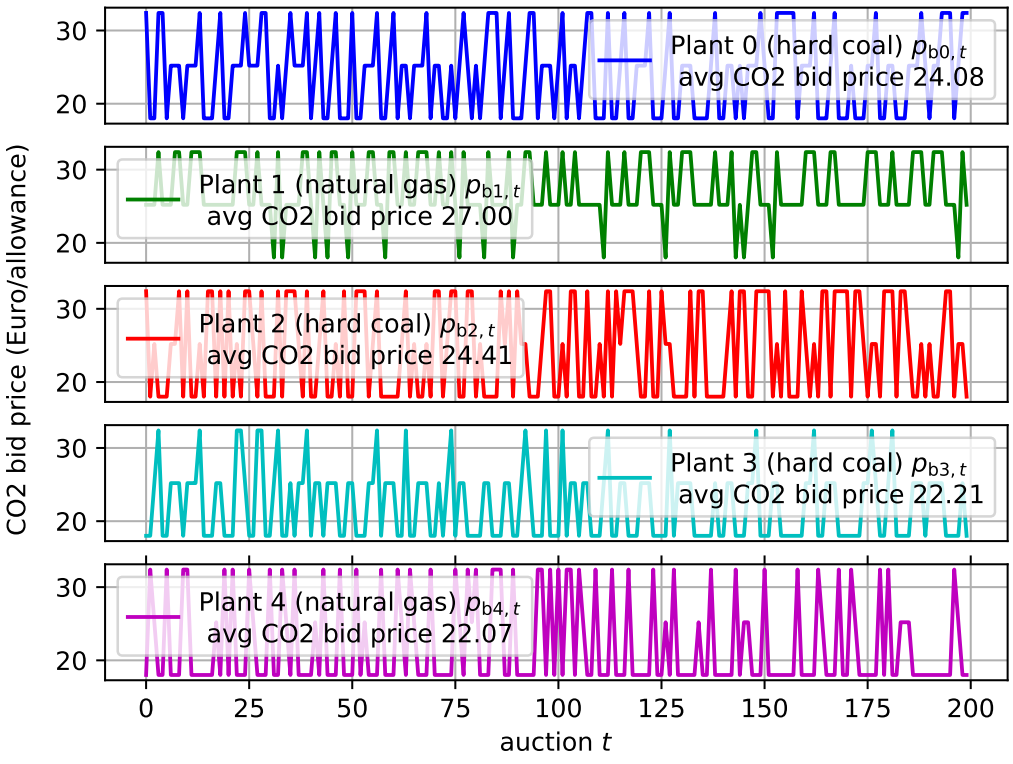


(b) Sarsa(λ).

Figure 5.10: Action entry: Carbon capture levels during testing.

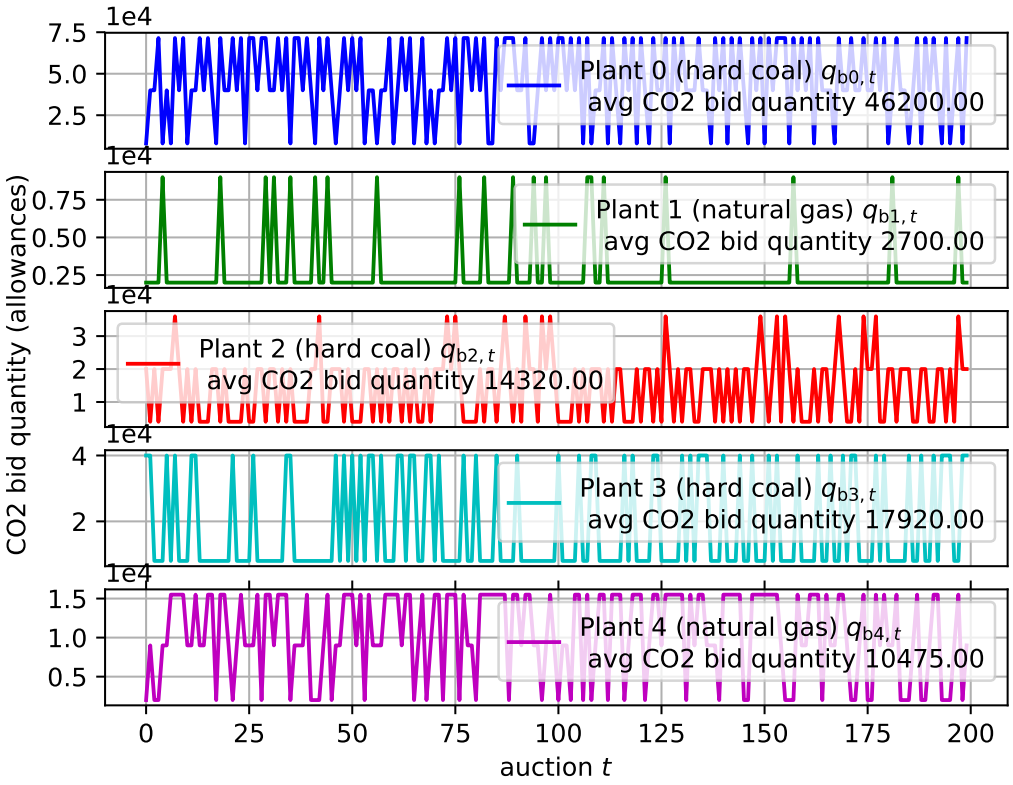


(a) Sarsa.

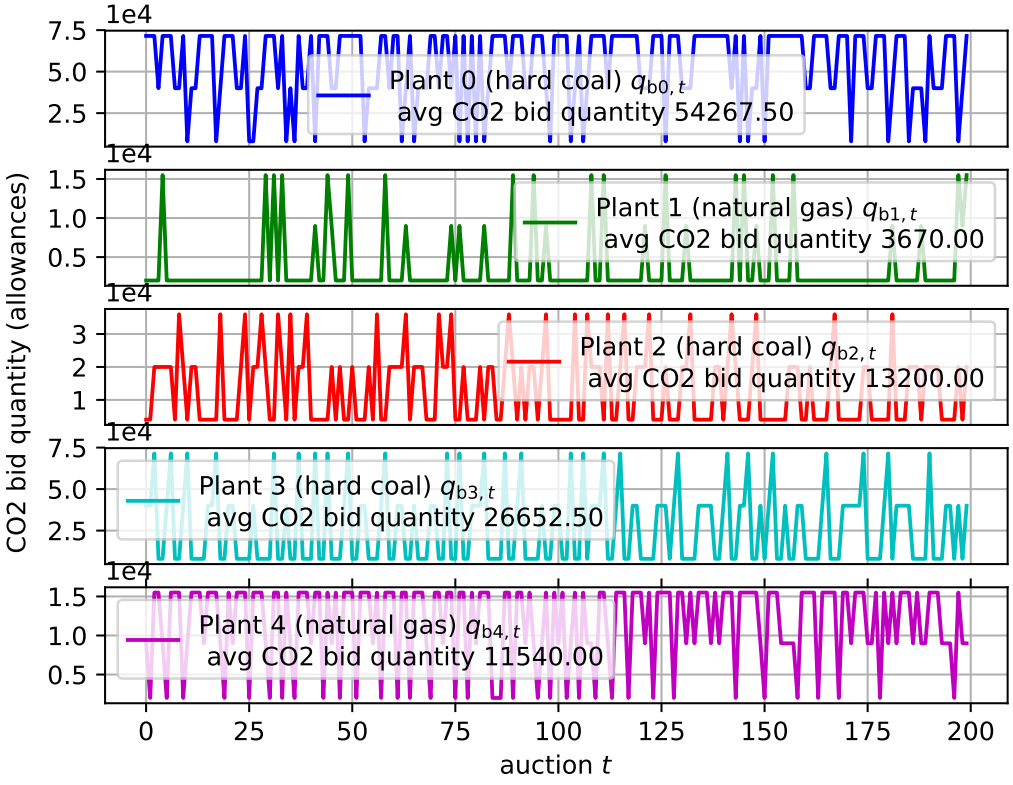


(b) Sarsa(λ).

Figure 5.11: Action entry: Bid prices during testing.

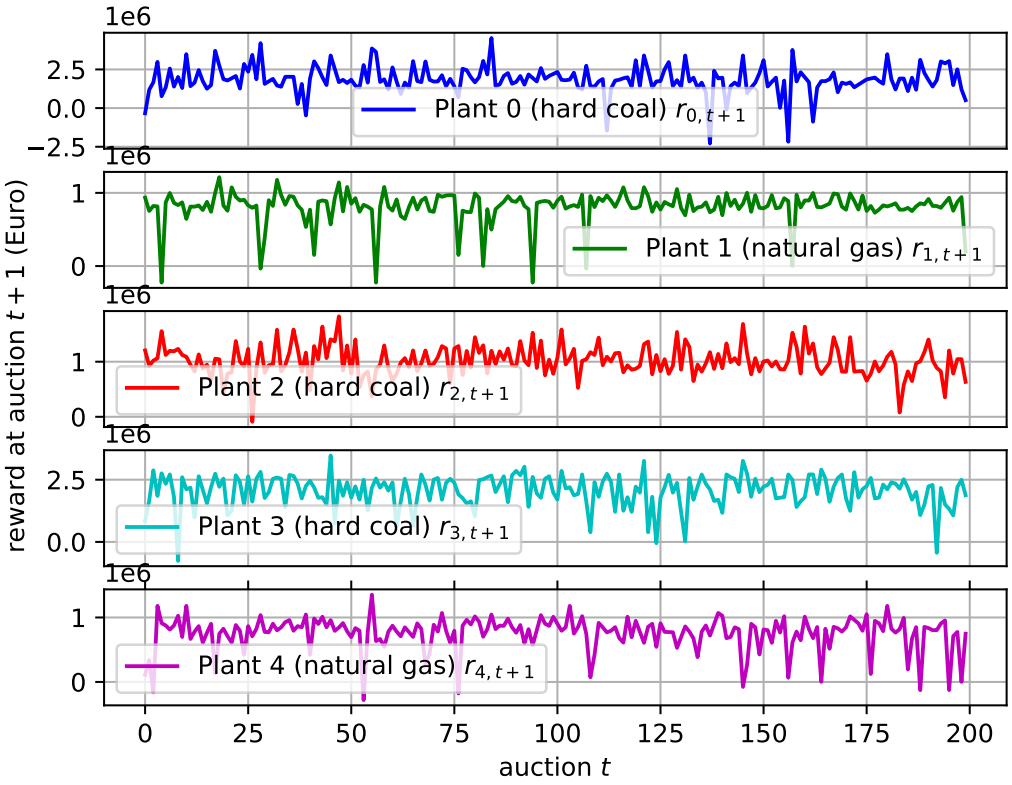


(a) Sarsa.

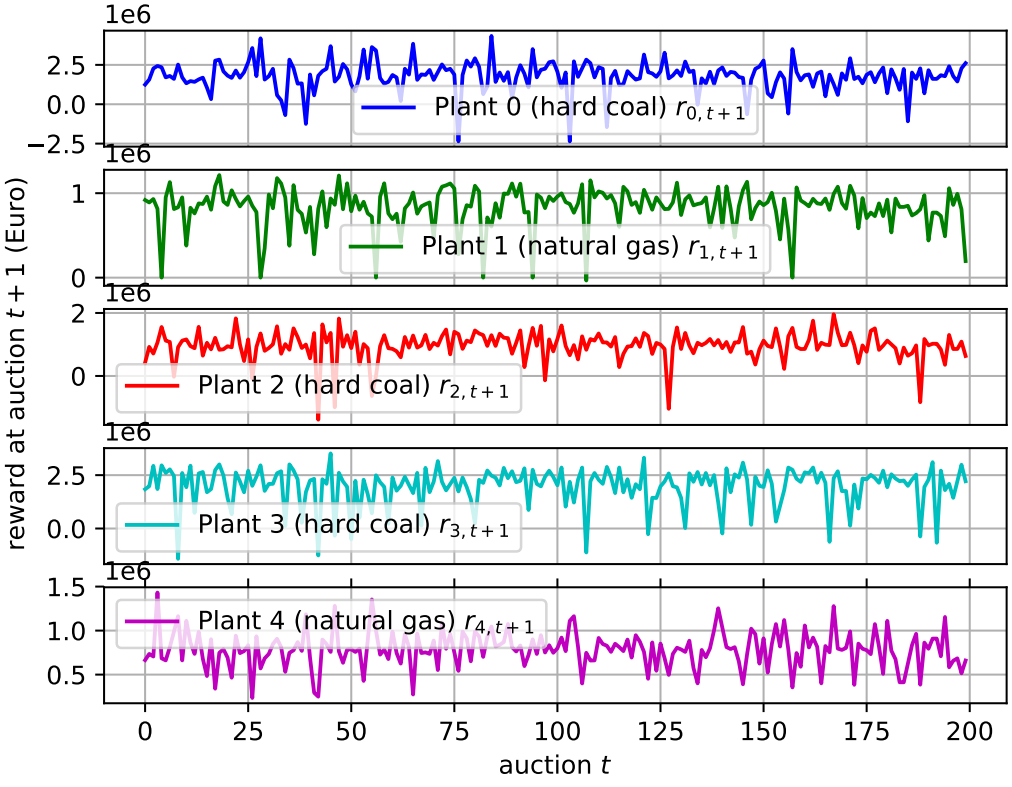


(b) Sarsa(λ).

Figure 5.12: Action entry: Bid quantities during testing.



(a) Sarsa.



(b) Sarsa(λ).

Figure 5.13: Rewards during testing.

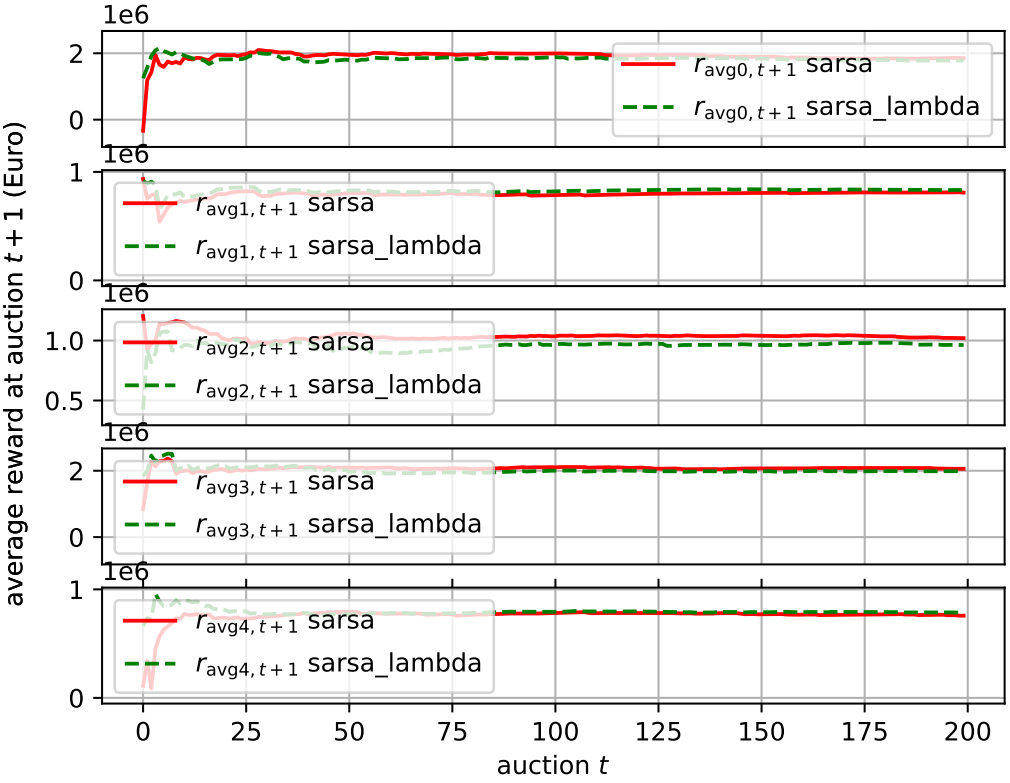


Figure 5.14: Average rewards during testing.

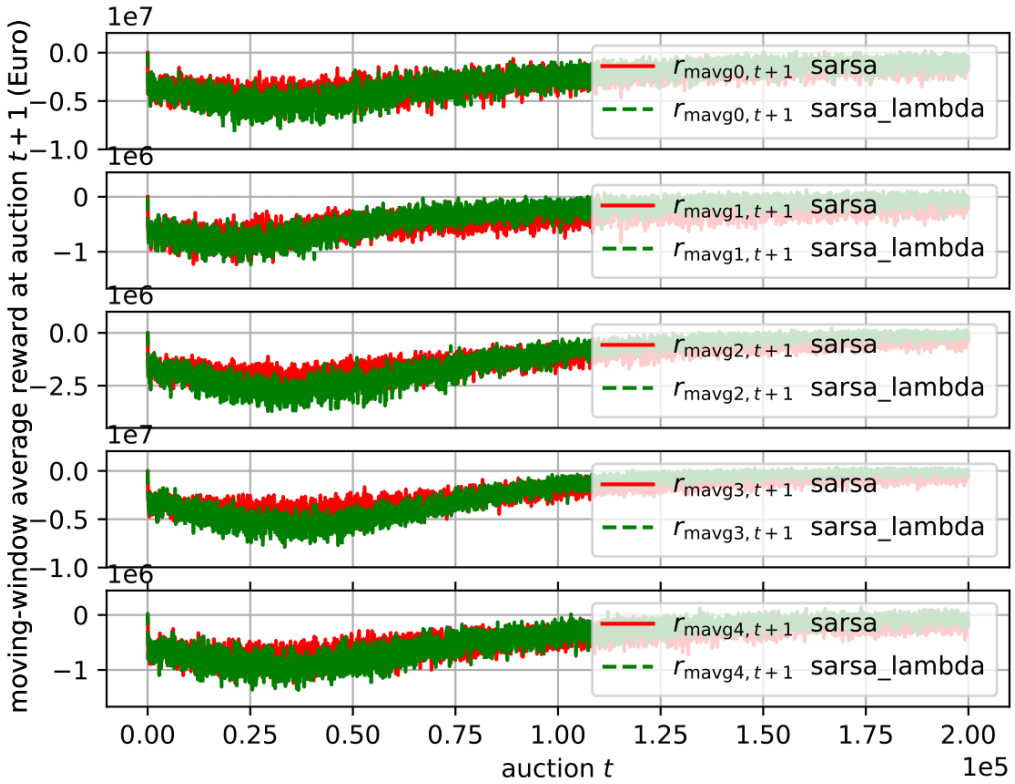


Figure 5.15: Moving-average rewards (every 100 steps) of Sarsa and Sarsa(λ) during training with $M = 3$.

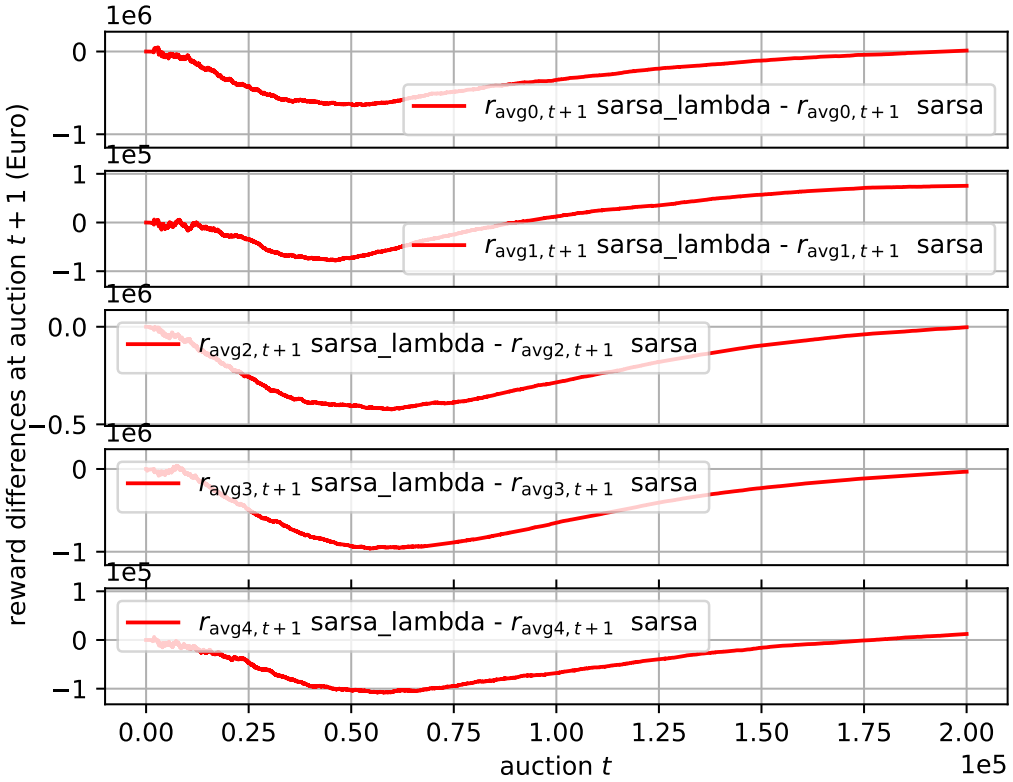


Figure 5.16: Average reward differences of Sarsa and Sarsa(λ) during training with $M = 3$.

Chapter 6

Conclusions

In this thesis, several important issues for the flexible operation of the MEA-based carbon capture plant integrated with a fossil-fuel power plant have been addressed within a unified model-free framework.

Since the MEA-based PCC process using the rate-based approach [43] is too complex for the model-based control, data generated by this first-principle model is used for the identification of a simpler neural-network-based model. The identified model can fit well with the data set and pass the residual analysis. Moreover, the GPC protocol has been implemented within this validated plant model (Chapter 2). As demonstrate in the simulations, GPC is a prevailing method which can be applied with only measured inputs and outputs data. Nevertheless, some offline model parameters such as $\mathbf{A}(q)$, $\mathbf{B}(q)$ and $\mathbf{D}(q)$ in the prediciton model (Eq. (3.40)) are required for the controller design, which makes the implementation of GPC non-trivial.

To solve the tracking issue of the carbon capture level, PI control, CFDL-MFAC and PFDL-MFAC protocols have been implemented on the identified plant model (Chapter 3). It is demonstrated that without external disturbances, both the CFDL- and PFDL-MFAC can perform similarly to GPC and PI control protocols. Thereafter, the fluctuations of flue gas flow rate and the mass fraction of CO_2 in the flue gas are introduced to test the robustness of the aforementioned controllers. These fluctuations serve as the external disturbances of the MEA-based PCC plant and degrade

the performances of the controllers. In a real environment, such disturbances may be caused by the associated fossil-fuel power plant with time-vary load conditions. It has been observed that the PFDL-MFAC should have more robust performances than the PI control and CFDL-MFAC. Furthermore, the PFDL-MFAC can achieve some dynamic responses similar to the GPC protocol. Nevertheless, the PFDL-MFAC requires no model parameters while the GPC necessitates an offline model for the online implementation, which implies PFDL-MFAC should be more favourable.

To determine the carbon capture level for the MEA-based PCC plants, Sarsa and Sarsa(λ) learning algorithms are applied to finding the optimal policies. These decision-making methods have taken into account the operation of associated fossil-fuel power plant, variations of the electricity market prices and bidding of CO_2 allowance auctions, which is seldom considered in the previous literature. Then, a unified bidding and operation problem of the MEA-based carbon capture plant is formulated in order to find the optimised carbon capture levels under the reinforcement learning framework. The relationship between the operation and bidding are established by holding accounts of the agents rather than instantly balancing between the allowances and the CO_2 emission. The performances of the two algorithms are compared with each other, which implies Sarsa should be preferred if the data for learning can be generated cheaply and there is no delay-delayed penalties. However, in some initial learning steps, Sarsa(λ) usually learns faster than Sarsa. When data are scarce and valuable or the environment involves some time-varying or delayed-penalty property, Sarsa(λ) could learn much faster.

Nevertheless, there still exist some future works of the flexible operation issues on the solvent-based PCC plants integrated with the fossil-fuel power plants, which are listed as follows.

- In this thesis, the robustness of MFAC is compared with PI and GPC when disturbances are applied. To deal with those disturbance directly, some disturbance compensation or rejection methods can be implemented.
- The model-free control protocol in this thesis only solves the carbon capture level tracking problem of the MEA-based PCC plant since currently, only limited data

are received from the collaborators [43]. Direct extensions to the MIMO cases can be done with the MIMO PFDL-MFAC (Eqs. (4.44), (4.45), (4.46) and (4.47)) or the MIMO CFDL-MFAC protocols (Eqs. (4.40), (4.41), (4.42) and (4.43)) if some more input-output data generated by the first-principle MEA-based PCC model can be acquired.

- There are some other model-free control protocols which can be possibly implemented for the preceding carbon capture plant, such as the reinforcement-leaning based linear quadratic regulator [90, 91, 92], which is both model-free and optimal.

- The flexible carbon capture levels are given by tabular learning algorithms (Tables 5.5 and 5.6) through state aggregation. Function approximators [36] can be applied which may potentially give more accurate and smooth estimations of the action value functions. In addition, there exist another set of reinforcement learning algorithms which introduce parametrisation of policies [37] and enable the selections of stochastic policies.

- The electricity price is specified as a simple profile (Figures 5.1) whereas it is actually determined by different electricity market conditions. Further researches may extend the electricity price in more practical ways. To make Sarsa or Sarsa(λ) applicable to these situations, some prediction methods such as auto-regressive moving-average model, neural networks, etc., may be used together with those learning algorithms.

Bibliography

[1] S. Alberici, S. Boeve, P. Van Breevoort, Y. Deng, S. Förster, A. Gardiner, V. van Gastel, K. Grave, H. Groenenberg, D. de Jager *et al.*, "Subsidies and costs of EU energy," Report, 2014.

[2] U. S. EIA, "Capital cost estimates for utility scale electricity generating plants," U.S. Energy Inf. Adm, Report, 2016.

[3] G. Manzolini, E. Sanchez Fernandez, S. Rezvani, E. Macchi, E. Goetheer, and T. Vlugt, "Economic assessment of novel amine based CO_2 capture technologies integrated in power plants based on European Benchmarking Task Force methodology," *Applied Energy*, vol. 138, pp. 546–558, 2015.

[4] B. Metz, O. Davidson, H. de Coninck, M. Loos, and L. Meyer, "IPCC special report on carbon dioxide capture and storage," pp. 5–10, 2005.

[5] E. Kenig, R. Schneider, and A. Grak, "Reactive absorption: Optimal process design via optimal modelling," *Chemical Engineering Science*, vol. 56, no. 2, pp. 343–350, 2001.

[6] Z. Li, Z. Ding, and M. Wang, "Optimal bidding and operation of a power plant with solvent-based carbon capture under a CO_2 allowance market: A solution with a reinforcement learning-based sarsa temporal-difference algorithm," *Engineering*, vol. 3, no. 2, pp. 257–265, 2017.

[7] F. Slye, "UK future energy scenarios: System operator," *National Grid UK*, 2018.

[8] M. van der Hoeven, "Technology roadmap carbon capture and storage 2013ed."

[9] M. Bui, C. S. Adjiman, A. Bardow, E. J. Anthony, A. Boston, S. Brown, P. S. Fennell, S. Fuss, A. Galindo, L. A. Hackett *et al.*, "Carbon capture and storage (CCS): the way forward," *Energy & Environmental Science*, vol. 11, no. 5, pp. 1062–1176, 2018.

[10] M. Bui, I. Gunawan, V. Verheyen, P. Feron, E. Meuleman, and S. Adeloju, "Dynamic modelling and optimisation of flexible operation in post-combustion CO_2 capture plants-A review," *Computers & Chemical Engineering*, vol. 61, no. Supplement C, pp. 245–265, 2014.

[11] M. Wang, A. Lawal, P. Stephenson, J. Sidders, and C. Ramshaw, "Post-combustion CO_2 capture with chemical absorption: A state-of-the-art review," *Chemical Engineering Research and Design*, vol. 89, no. 9, pp. 1609–1624, 2011.

[12] J. Black *et al.*, "Cost and performance baseline for fossil energy plants Volume 1: Bituminous coal and natural gas to electricity," *National Energy Technology Laboratory: Washington, DC, USA*, 2010.

[13] S. Reddy, J. Scherffius, S. Freguia, and F. Fluor's Econamine, "Plussm technology: An enhanced amine-based CO_2 capture process," in *Second National Conference on Carbon Sequestration National Energy Technology Laboratory/Department of Energy, Alexandria, VA, May*, 2003, pp. 5–8.

[14] A. Lawal, M. Wang, P. Stephenson, and O. Obi, "Demonstrating full-scale post-combustion CO_2 capture for coal-fired power plants through dynamic modelling and simulation," *Fuel*, vol. 101, no. Supplement C, pp. 115–128, 2012.

[15] Z. Li, Z. Ding, and M. Wang, "Operation and bidding strategies of power plants with carbon capture," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3244–3249, 2017, 20th IFAC World Congress.

[16] S. M. Cohen, G. T. Rochelle, and M. E. Webber, "Optimizing post-combustion CO_2 capture in response to volatile electricity prices," *International Journal of Greenhouse Gas Control*, vol. 8, pp. 180–195, 2012.

[17] X. Luo and M. Wang, "Optimal operation of MEA-based post-combustion carbon capture for natural gas combined cycle power plants under different market

conditions," *International Journal of Greenhouse Gas Control*, vol. 48, pp. 312–320, 2016.

[18] "California cap on greenhouse gas emissions and market-based compliance mechanisms," *17 C.C.R. § 95801-96022*, 2018.

[19] "ICE Report Center," https://www.theice.com/marketdata/reports/148, 2018, accessed: 2018-07-03.

[20] T. Nittaya, P. L. Douglas, E. Croiset, and L. A. Ricardez-Sandoval, "Dynamic modelling and control of MEA absorption processes for CO_2 capture from power plants," *Fuel*, vol. 116, pp. 672–691, 2014.

[21] Y. J. Lin, T. H. Pan, D. S. H. Wong, S. S. Jang, Y. W. Chi, and C. H. Yeh, "Plantwide control of CO_2 capture by absorption and stripping using monoethanolamine solution," *Industrial & Engineering Chemistry Research*, vol. 50, no. 3, pp. 1338–1345, 2010.

[22] G. Léonard, B. C. Mogador, S. Belletante, and G. Heyen, "Dynamic modelling and control of a pilot plant for post-combustion CO_2 capture," in *Computer Aided Chemical Engineering*, 2013, vol. 32, pp. 451–456.

[23] B. W. Bequette, *Process control: modeling, design, and simulation.* Prentice Hall Professional, 2003.

[24] M. H. Sahraei and L. Ricardez-Sandoval, "Controllability and optimal scheduling of a CO_2 capture plant using model predictive control," *International Journal of Greenhouse Gas Control*, vol. 30, pp. 58–71, 2014.

[25] E. F. Camacho and C. B. Alba, *Model predictive control.* Springer Science & Business Media, 2013.

[26] W. H. Kwon and S. H. Han, *Receding horizon control: model predictive control for state models.* Springer Science & Business Media, 2006.

[27] Z. Hou, H. Gao, and F. L. Lewis, "Data-driven control and learning systems," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4070–4075, 2017.

[28] M. Panahi and S. Skogestad, "Economically efficient operation of CO_2 capturing process. Part II. Design of control layer," *Chemical Engineering and Processing: Process Intensification*, vol. 52, pp. 112 – 124, 2012.

[29] Z. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.

[30] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of MIMO nonlinear discrete-time systems," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2173–2188, 2011.

[31] ——, "A novel data-driven control approach for a class of discrete-time nonlinear systems," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 6, pp. 1549–1558, 2011.

[32] Z. Li, Z. Ding, M. Wang, and E. Oko, "Model-free adaptive control for mea-based post-combustion carbon capture processes," *Fuel*, vol. 224, pp. 637 – 643, 2018.

[33] H. Chalmers, M. Lucquiaud, J. Gibbins, and M. Leach, "Flexible operation of coal fired power plants with postcombustion capture of carbon dioxide," *Journal of Environmental Engineering*, vol. 135, no. 6, pp. 449–458, 2009.

[34] M. Panahi and S. Skogestad, "Economically efficient operation of CO_2 capturing process part I: Self-optimizing procedure for selecting the best controlled variables," *Chemical Engineering and Processing: Process Intensification*, vol. 50, no. 3, pp. 247–253, 2011.

[35] N. Mac Dowell and N. Shah, "Identification of the cost-optimal degree of CO_2 capture: an optimisation study using dynamic process models," *International Journal of Greenhouse Gas Control*, vol. 13, no. Supplement C, pp. 44–58, 2013.

[36] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2010, vol. 39.

[37] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 2nd ed. MIT press, 2018.

[38] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[39] M. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, p. 3, 2012.

[40] M. T. Luu, N. A. Manaf, and A. Abbas, "Dynamic modelling and control strategies for flexible operation of amine-based post-combustion CO_2 capture systems," *International Journal of Greenhouse Gas Control*, vol. 39, no. Supplement C, pp. 377–389, 2015.

[41] X. Luo, M. Wang, E. Oko, and C. Okezue, "Simulation-based techno-economic evaluation for optimal design of CO_2 transport pipeline network," *Applied Energy*, vol. 132, pp. 610–620, 2014.

[42] R. E. Dugas, "Pilot plant study of carbon dioxide capture by aqueous monoethanolamine," Ph.D. dissertation, 2006.

[43] C. Biliyok, A. Lawal, M. Wang, and F. Seibert, "Dynamic modelling, validation and analysis of post-combustion chemical absorption CO_2 capture plant," *International Journal of Greenhouse Gas Control*, vol. 9, pp. 428–445, 2012.

[44] A. Lawal, M. Wang, P. L. Stephenson, and H. Yeung, "Dynamic modelling of CO_2 absorption for post combustion capture in coal-fired power plants," *Fuel*, vol. 88, no. 12, pp. 2455–2462, 2009.

[45] E. Oko, M. Wang, and A. K. Olaleye, "Simplification of detailed rate-based model of post-combustion CO_2 capture for full chain CCS integration studies," *Fuel*, vol. 142, pp. 87 – 93, 2015.

[46] K. R. Westerterp, W. Van Swaaij, and A. Beenackers, *Chemical reactor design and operation*. Wiley Chichester, 1987, vol. 84.

[47] A. Plus, "Rate based model of the CO_2 capture process by MEA using Aspen Plus," *Aspen Technology Inc, Cambridge, MA, USA*, 2008.

[48] A. Lawal, M. Wang, P. Stephenson, and H. Yeung, "Dynamic modeling and simulation of CO_2 chemical absorption process for coal-fired power plants," in *Computer Aided Chemical Engineering*. Elsevier, 2009, vol. 27, pp. 1725–1730.

[49] N. Harun, T. Nittaya, P. L. Douglas, E. Croiset, and L. A. Ricardez-Sandoval, "Dynamic simulation of MEA absorption process for CO_2 capture from power plants," *International Journal of Greenhouse Gas Control*, vol. 10, no. Supplement C, pp. 295–309, 2012.

[50] J. Åkesson, C. D. Laird, G. Lavedan, K. Prölß, H. Tummescheit, S. Velut, and Y. Zhu, "Nonlinear model predictive control of a CO_2 post-combustion absorption unit," *Chemical Engineering & Technology*, vol. 35, no. 3, pp. 445–454, 2012.

[51] A. Arce, N. Mac Dowell, N. Shah, and L. F. Vega, "Flexible operation of solvent regeneration systems for CO_2 capture processes using advanced control techniques: Towards operational cost minimisation," *International Journal of Greenhouse Gas Control*, vol. 11, no. Complete, pp. 236–250, 2012.

[52] N. Sipcz, F. A. Tobiesen, and M. Assadi, "The use of artificial neural network models for CO_2 capture plants," *Applied Energy*, vol. 88, no. 7, pp. 2368–2376, 2011.

[53] F. Li, J. Zhang, E. Oko, and M. Wang, "Modelling of a post-combustion CO_2 capture process using neural networks," *Fuel*, vol. 151, no. Supplement C, pp. 156–163, 2015.

[54] Y.-J. Lin, D. S.-H. Wong, S.-S. Jang, and J.-J. Ou, "Control strategies for flexible operation of power plant with CO_2 capture plant," *AIChE Journal*, vol. 58, no. 9, pp. 2697–2704, 2012.

[55] E. Mechleri, A. Lawal, A. Ramos, J. Davison, and N. M. Dowell, "Process control strategies for flexible operation of post-combustion CO_2 capture plants," *International Journal of Greenhouse Gas Control*, vol. 57, pp. 14 – 25, 2017.

[56] Z. He, M. H. Sahraei, and L. A. Ricardez-Sandoval, "Flexible operation and simultaneous scheduling and control of a CO_2 capture plant using model predictive

control," *International Journal of Greenhouse Gas Control*, vol. 48, pp. 300 – 311, 2016.

[57] X. Wu, M. Wang, J. Shen, Y. Li, A. Lawal, and K. Y. Lee, "Reinforced coordinated control of coal-fired power plant retrofitted with solvent based CO_2 capture using model predictive controls," *Applied Energy*, vol. 238, pp. 495 – 515, 2019.

[58] K. Prölß, H. Tummescheit, S. Velut, and J. Åkesson, "Dynamic model of a post-combustion absorption unit for use in a non-linear model predictive control scheme," *Energy Procedia*, vol. 4, pp. 2620 – 2627, 2011, 10th International Conference on Greenhouse Gas Control Technologies.

[59] J. Peng, T. F. Edgar, and R. B. Eldridge, "Dynamic rate-based and equilibrium models for a packed reactive distillation column," *Chemical Engineering Science*, vol. 58, no. 12, pp. 2671–2680, 2003.

[60] L. Ljung, *System Identification: Theory for the user*. PTR Prentice Hall Information and System Sciences Series, 1987.

[61] S. Ziaii, G. T. Rochelle, and T. F. Edgar, "Dynamic modeling to minimize energy use for CO_2 capture in power plants by aqueous monoethanolamine," *Industrial & Engineering Chemistry Research*, vol. 48, no. 13, pp. 6105–6111, 2009.

[62] S. Ziaii, S. Cohen, G. T. Rochelle, T. F. Edgar, and M. E. Webber, "Dynamic operation of amine scrubbing in response to electricity demand and pricing," *Energy Procedia*, vol. 1, no. 1, pp. 4047–4053, 2009.

[63] A. Lawal, M. Wang, P. L. Stephenson, G. C. Koumpouras, and H. Yeung, "Dynamic modelling and analysis of post-combustion CO_2 chemical absorption process for coal-fired power plants," *Fuel*, vol. 89, no. 10, pp. 2791–2801, 2010.

[64] M. R. Abu-Zahra, J. P. Niederer, P. H. Feron, and G. F. Versteeg, "CO_2 capture from power plants: Part II. A parametric study of the economical performance based on mono-ethanolamine," *International Journal of Greenhouse Gas Control*, vol. 1, no. 2, pp. 135–142, 2007.

[65] E. Agbonghae, K. Hughes, D. Ingham, L. Ma, and M. Pourkashanian, "Optimal process design of commercial-scale amine-based CO_2 capture plants," *Industrial & Engineering Chemistry Research*, vol. 53, no. 38, pp. 14 815–14 829, 2014.

[66] A. Galán-Martín, C. Pozo, A. Azapagic, I. Grossmann, N. Mac Dowell, and G. Guillén-Gosálbez, "Time for global action: an optimised cooperative approach towards effective climate change mitigation," *Energy & Environmental Science*, vol. 11, no. 3, pp. 572–581, 2018.

[67] M. R. Abu-Zahra, L. H. Schneiders, J. P. Niederer, P. H. Feron, and G. F. Versteeg, "CO_2 capture from power plants: Part I. A parametric study of the technical performance based on monoethanolamine," *International Journal of Greenhouse gas control*, vol. 1, no. 1, pp. 37–46, 2007.

[68] A. Aroonwilas and A. Veawab, "Integration of CO_2 capture unit using single- and blended-amines into supercritical coal-fired power plants: Implications for emission and energy management," *International Journal of Greenhouse Gas Control*, vol. 1, no. 2, pp. 143 – 150, 2007, 8th International Conference on Greenhouse Gas Control Technologies.

[69] C. Biliyok and H. Yeung, "Evaluation of natural gas combined cycle power plant for post-combustion CO_2 capture integration," *International Journal of Greenhouse Gas Control*, vol. 19, pp. 396 – 405, 2013.

[70] P. Luckow, E. A. Stanton, S. Fields, B. Biewald, S. Jackson, J. Fisher, and R. Wilson, "2015 carbon dioxide price forecast," *Cambridge, MA: Synapse Energy Economics, Inc*, vol. 3, 2015.

[71] Y. Chen and L. Wang, "A power market model with renewable portfolio standards, green pricing and GHG emissions trading programs," in *Energy 2030 Conference, 2008. ENERGY 2008. IEEE.* IEEE, 2008, pp. 1–7.

[72] V. Nanduri, "Application of reinforcement learning-based algorithms in CO_2 allowance and electricity markets," in *2011 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL).* IEEE, 2011, pp. 164–169.

[73] "EU Emissions Trading System (EU ETS)," https://ec.europa.eu/clima/policies/ets_en, 2018, accessed: 2019-03-22.

[74] K. P. Burnham and D. R. Anderson, *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2002.

[75] H. J. Larson and H. J. Larson, *Introduction to probability theory and statistical inference*. Wiley New York, 1969, vol. 12.

[76] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[77] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[78] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[79] D. Clarke, C. Mohtadi, and P. Tuffs, "Generalized predictive control-Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[80] ——, "Generalized predictive control-Part II. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149 – 160, 1987.

[81] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*. Springer, 1999, pp. 207–226.

[82] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[83] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Courier Corporation, 2014.

[84] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," in *Innovations in multi-agent systems and applications-1*. Springer, 2010, pp. 183–221.

[85] S. M. Ross, *Introduction to stochastic dynamic programming.* Academic press, 2014.

[86] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming.* Athena Scientific Belmont, MA, 1996, vol. 5.

[87] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, "Convergence results for single-step on-policy reinforcement-learning algorithms," *Machine Learning*, vol. 38, no. 3, pp. 287–308, Mar 2000.

[88] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems.* Springer Science & Business Media, 2012, vol. 26.

[89] E. S. Fernandez, E. Goetheer, G. Manzolini, E. Macchi, S. Rezvani, and T. Vlugt, "Thermodynamic assessment of amine based CO_2 capture technologies in power plants based on European Benchmarking Task Force methodology," *Fuel*, vol. 129, pp. 318–329, 2014.

[90] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[91] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE Transactions on Cybernetics*, vol. 45, no. 12, pp. 2770–2779, 2015.

[92] B. Kiumarsi, F. L. Lewis, and Z.-P. Jiang, "H∞ control of linear discrete-time systems: Off-policy reinforcement learning," *Automatica*, vol. 78, pp. 144 – 152, 2017.