



Einberger Bernhard, BSc.

Machine Learning Model to Predict Fuel Cell Degradation Effects

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Mechanical Engineering and Business Economics

submitted to

Graz University of Technology

Supervisor

Univ.-Prof Dr. Grohs Philipp

Institute of Geometry

Head: Univ.-Prof Dr. Wallner Johannes

Graz, August 2023

This document is set in Palatino, compiled with pdfL^AT_EX2e and Biber.

The L^AT_EX template from Karl Voit is based on KOMA script and can be found online: <https://github.com/novoid/LaTeX-KOMA-template>

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

Fuel Cell technology could take a key role in the industries transformation towards decarbonization. Therefore, the component must be well designed and long-lasting.

Several simulation tools are developed by AVL in order to increase the engineering capability on that technology.

This thesis introduces a workflow that describes how to generate Fuel Cell degradation data with a commercial 3D-CFD solver (FireM[®]) and process the data by means of Machine Learning tools.

A detailed simulation model is utilized to calibrate a simplified simulation model. The Electrocatalytic Surface Area suggested by the simplified simulation model differs with measurements by roughly 30%.

Two ML models were set up to predict degradation effects, a Kernel Ridge Regression, and a Dense Neural Network model. The following degradation effects are predicted: current density reduction, equivalent weight (membrane), platinum particle number (CAT-CL), specific platinum surface (CAT-CL), membrane thickness, CAT-CL thickness and ANO-CL thickness.

The ML model's prediction error is less than 4% for each degradation effect.

A way to de-feature time series data (e.g., load profile) with utilizing Principal Component Analysis is introduced and proven to be applicable at this task.

The purpose of the ML model is to better understand the effect of operating conditions onto Fuel Cell degradation and therefore increase the engineering capability.

Contents

Abstract	v
Symbols, Constants and Abbreviations	xi
1 Introduction	1
2 Theoretical Part	3
2.1 Fuel Cell	3
2.1.1 Operating Principle	3
2.1.2 Polymer Electrolyte Membrane FC	8
2.2 Multiphysics CFD Solver	10
2.2.1 Navier-Stokes equations	10
2.2.2 Electromagnetic	12
2.2.3 FC-Balancing Equations	13
2.2.4 FC-Degradation Modelling	16
2.3 Machine Learning	30
2.3.1 Principal Component Analysis	30
2.3.2 Kernel Ridge Regression	33
2.3.3 Dense Neural Network	38
3 Practical Part	41
3.1 CFD Simulation Model	41
3.1.1 Analysis Model	42
3.1.2 Boundary Conditions	46
3.1.3 Numerical Setup	49
3.1.4 Model Calibration	52
3.2 Machine Learning Model	56
3.2.1 Data Preprocessing	56
3.2.2 Dataset/Training Strategy	61

Contents

3.2.3	ML Model Setup	64
4	Results	69
4.1	Comparison detailed vs. homogenized simulation	69
4.2	Comparison simulation vs. measurement	74
4.3	Homogenized model	75
4.4	Machine learning model comparison	77
4.5	KRR - prediction trend analysis	83
5	Outlook	85
	Bibliography	87

List of Figures

2.1	Simple fuel cell [11]	4
2.2	Schematic of hydrogen oxidation reaction [11]	5
2.3	Polarization curve [11]	7
2.4	Schematic assembly of a fuel cell stack [17]	8
2.5	Schematic of H ₂ , O ₂ PEM fuel cell[11]	9
2.6	PCA subspace decision visualization [4]	31
2.7	Transformation input feature space to intrinsic space [8]	35
2.8	Visualization of a perceptron or neuron [4]	38
2.9	Neural network [4]	40
3.1	Detailed model	43
3.2	Homogenized model	45
3.3	Simulation load cycle	50
3.4	Calibrated polarization curve - detailed model	54
3.5	Calibrated polarization curve - homogenized model	54
3.6	Calibrated pressure drop - anode (left) and cathode (right) . .	56
3.7	Example of the class approach decomposition	58
3.8	Example of trapezoidal curve data	59
3.9	Example of triangular curve data	59
3.10	Example of piecewise constant curve data	60
3.11	Initial Labels	62
3.12	Transformed Labels	63
3.13	Dataset strategy	65
4.1	Comparison cathode flow: Velocity - Total Pressure - Temperature	70
4.2	Comparison anode flow: Velocity - Total Pressure - Temperature	70

List of Figures

4.3	Comparison membrane: Ionic current density - dissolved water flux - dissolved water concentration	71
4.4	Comparison coolant flow: Velocity - Absolute Pressure	72
4.5	Comparison coolant flow: Temperature on cathode and anode side	72
4.6	Comparison cathode catalyst layer: Ionic current density - equilibrium potential	73
4.7	ECSA loss Comparison: simulation vs. measurement	74
4.8	Degradation trends on the homogenized mode	75
4.9	2D Results of the homogenized model	76
4.10	MAPE - Class approach	77
4.11	MAPE - PCA approach	78
4.12	MAPE - PCA adapted approach	79
4.13	MAPE distribution - KRR - Class approach	80
4.14	MAPE distribution - DNN - Class approach	80
4.15	MAPE distribution - KRR - PCA adapted approach	81
4.16	MAPE distribution - DNN - PCA adapted approach	81
4.17	Summary of prediction accuracy	82
4.18	Trend analysis	83

Symbols, Constants and Abbreviations

Latin symbols

a_C	1/m	Carbon surface density
a_{Pt}	1/m	Platinum surface density
A_C	m ² /kg	specific carbon surface
A_{Pt}	m ² /kg	specific platinum surface
c	mol/m ³	Molar concentration
C_w	-	Dissolved water content
d_{cc}	m	Carbon corrosion depth
D	m ² /s	Diffusion coefficient
EW	kg/mol	Equivalent weight of ionomer
e	J/m ³	Total internal energy per volume
f_C	-	Free carbon surface fraction
f_{CO}	-	C=O covered carbon surface
f_{COH}	-	C-OH covered carbon surface
f_{Pt}	-	Free platinum surface fraction
f_{PtO}	-	Pt-O covered platinum surface
f_{PtH}	-	Pt-OH covered platinum surface
G	J	Gibbs free energy
H	J	Enthalpy
i	A/m ²	Current density
J	A	Current
k_{det}	-	Detachment rate
k	-	Constant reaction rate
m	kg	Mass
M	kg/mol	Molar mass
$n_{Pt,i}$	1/m ³	Platinum particle number density

Symbols and abbreviations

Q	J/m ³	Volumetric heat
\vec{q}	W	Heat conduction
r	m	Radius
r_{OX}	-	Potential drop across Pt oxide film
R	J/kgK	Individual gas constant
S	J/K	Entropy
S	-	Source term for a balance equation
t	s	Time
T	K	Temperature
$x_{k,j}$	—	Mole fraction k of phase j
$x_{k,j}$	—	Mass fraction k of phase j
z	—	Amount of charge/electrons

Greek Symbols

α	-	Phase volume fraction
β	-	Charge transfer coefficient
δ	m	Thickness
ϵ	-	Porosity
η_{act}	V	Activation potential
γ	kg/s ²	Surface tension
ϕ	V	Potential
$\phi_{eq,0}$	V	Standard equilibrium potential
$\phi_{eq,0}^0$	V	Standard equilibrium potential at norm conditions
ϕ_{eq}	V	Equilibrium potential
ψ_{dec}	-	Tuning Parameter - Carbon Corrosion
τ	N/m ²	Shear stress
φ	%	Relative humidity

Constants

e	2.71828182846	-	Euler constant
e	$1.602E^{-19}$	C	Elemental charge

F	96485.3321	sA/mol	Faraday constant
R_m	8.314	J/molK	Universal gas constant
π	3.1415	-	Pi

Indices and Abbreviations

α	First stage
β	Second stage
o	Standard
act	Activation
agg	Agglomerated
ANO	Anode
att	Attached
bol	Begin of life
CAT	Cathode
cc	Carbon corrosion
cl	Catalyst layer
CL	Catalyst layer
CFD	Computational fluid dynamics
CPU	Central processing unit
det	Detached
dec	Deceleration
ECSA	Electrochemical surface area
ec	from electrochemical reaction
ele	Electronic phase
eq	Equilibrium
f	Forward reaction
FC	Fuel cell
g	Gaseous state
H(x)	Heaviside step function of x
ion	Ionic phase
id	Ionomer degradation
m	Molar
mer	Merged
ML	Machine Learning
no	Multiple number of main chain segments

Symbols and abbreviations

ohm	from ohmic resistance
par	from parasitic reactions
pb	from parasitic reactions
PCA	Principal component analysis
pd	Platinum dissolution
pt₂₊	Platinum ion
pc	from phase change
PEM	Polymer electrolyte membrane
RHE	Reversible hydrogen electrode
ref	Reference
R	Reaction
R_f	Placeholder for a larger molecule
s	Solid
SO	Solid oxide
SVD	Singular value decomposition
tot	Total
w	Within dissolved water

Please note that chemical symbols are not listed in the indices and abbreviations list.

1 Introduction

AVL List GmbH is researching Fuel Cells because they see it as a crucial component, necessary to decarbonize the industry in several fields of application, such as decentralized energy production, heavy transportation (trucks, planes), maritime sector or steel industry. AVL is approached frequently by customers who want to model the ageing effects on their fuel cells. Therefore, several simulation tools were developed in order to fulfil the demand. One simulation tool that has implemented Fuel Cell degradation models is AVL FireM. FireM is a commercial CFD-solver that has several Multi-physics modules implemented.

The motivation for this thesis arises out of two aspects. The first one is to test the degradation modelling capabilities, within FireM, to generate knowledge and expertise on that engineering task. The second factor is the idea to generate a dataset, with that simulation tool, and utilizing the data to train a Machine Learning model. The benefit of the Machine Learning workflow is the ability to produce fast predictions. The simulation model provides three dimensional results, which can be utilized for engineering, but they are computationally costly. One simulation on a simplified model takes up to ten days on six CPU's.

The thesis provides an overview of the main theory underlying both the simulation models and the Machine Learning models in chapter 2. Afterwards the model architecture and boundary conditions are described in chapter 3. In the results chapter 4 an overview of the simulation results and the Machine Learning results is given. Finally an outlook is presented in chapter 5.

2 Theoretical Part

In the theoretical part the most relevant theory, on which this work is based, is described. The chapter is split into three subchapters. The Fuel Cell subchapter (2.1) describes the basics of PEM Fuel Cells. In the Multi-physics CFD solver part (2.2), an overview on the simulation procedure and on the solved equations is given. The Machine Learning subchapter (2.3) describes the ML model basics and the dimension reduction method applied on the load profiles of the Fuel Cells.

2.1 Fuel Cell

This chapter gives an introduction into the operating principle (2.1.1) of a Fuel Cell. The cell type investigated in the practical part (3) is a PEM Fuel Cell. Therefore an introduction in PEM Fuel Cells is presented (2.1.2).

2.1.1 Operating Principle

This chapter provides information on Fuel Cell's with a focus on the conversion from chemical to electrical energy, its most relevant types, characteristics and components.

Fuel Cells convert chemical energy into electrical energy and heat. To do so, hydrogen oxidation takes place. The oxidation is performed by splitting the reaction into two half reactions. This is necessary to utilize the electrons that are involved in the reaction. The half reactions, equation 2.1 & 2.2, are spatially separated to transfer the electrons through an external circuit; this is shown in Figure 2.1 [11]

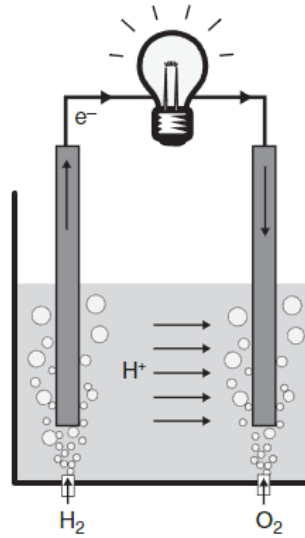


Figure 2.1: Simple fuel cell, invented by William Grove in 1839 [11]



Energy is transferred by the hydrogen oxidation, due to the lower energetic level of the bounding configuration of water compared to oxygen and hydrogen gas (figure 2.2) [11].

To spatially separate the half reactions, an electrolyte is utilized to create an environment in which only ions can be transferred, but no electrons. Two electrodes are inserted in the electrolyte on which the two half reactions take place. The Fuel Cell in Figure 2.1 consists of two platinum electrodes dipped into sulphonic acid. Sulphonic acid is an electrolyte which only enables H^+ ions to be transferred. [11]

Hydrogen and oxygen gas is fed to the assembly underneath the electrodes. The electrode on which electrons are removed from a species is called anode

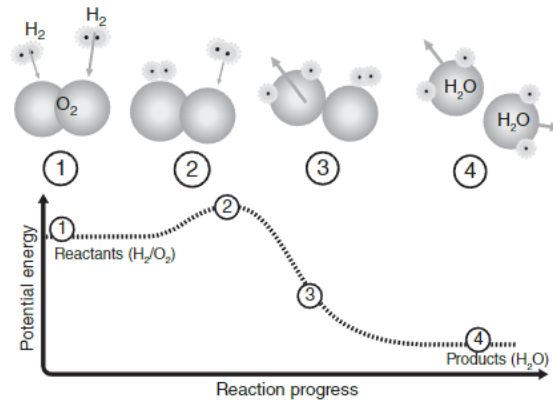


Figure 2.2: Schematic of hydrogen oxidation reaction. (1) hydrogen and oxygen gas, (2) breakup of bonds, requires energy, (3/4) formation of hydrogen-oxygen bounds leads to energy output [11]

and the process is known as oxidation. The electrode where electrons are added to an species is called cathode and the process is known as reduction. In Figure 2.1 the left electrode is the anode and the right one is the cathode. [11]

Fuel cells can be built up by several materials and characteristics, which results in different types. The two most popular types are SO-FC and PEM-FC. A overview of their characteristics is presented in table 2.1 [11].

The main performance measure of a fuel cell is the polarization curve. It shows the possible operating points, which are a pair of voltage and current. The current is usually normalized by the area to make different sizes comparable. [11]

The theoretical thermodynamic voltage produced in a fuel cell at norm conditions (E^0) is calculated by the free Gibbs energy difference due to the electrochemical reaction (ΔG_R^0), the Faraday constant (F) and the amount of involved electrons (n). Hydrogen oxidation has a maximal theoretical potential of +1.23 [V]. [11]

	PEMFC	SOFC
Electrolyte	Polymer membrane	Ceramic
Charge carrier	H ⁺	O ²⁻
Operating temperature	80°C	600-1000°C
Catalyst	Platinum	Perovskites
Cell components	Carbon based	Ceramic based
Fuel compatibility	H ₂ , methanol	H ₂ , CH ₄

Table 2.1: Fuel Cell Types [11]

$$E^0 = -\frac{\Delta G_R^0}{nF} [11] \quad (2.3)$$

There are three main losses (see Figure 2.5) which decrease the theoretical possible voltage described in equation 2.3 [11]:

- Activation losses
- Ohmic losses
- Concentration losses

To start the chemical reaction, an overpotential has to be supplied. This decreases the theoretical potential in order to start the reaction and is the main loss at low current densities. In order to transfer electrical current through a conductor, a potential difference is needed. The Ohmic law describes the potential loss due to conduction through a material. This loss is dominant at medium current densities. At high current densities lots of products are produced by the chemical reaction. These products must be transported downstream the fuel cell, but due to the big amount, it becomes more difficult to get rid of the products. This results in an increased concentration of products and a decreased concentration of reactants and furthermore into the concentration losses. All losses will decrease efficiency either by directly producing heat or causing reactants (H₂) to go through the system without taking part on the chemical reaction. [17]

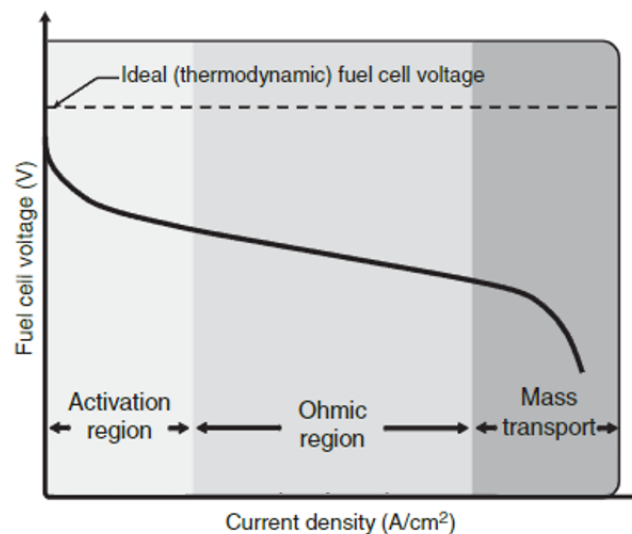


Figure 2.3: Polarization curve [11]

To increase the voltage output of a fuel cell, several layers are stacked upon each other to connect multiple Fuel Cells in serial. Figure 2.4 shows a fuel cell stack. It consists of the following components:

- End plates
- Bipolar plates
- MEA
- Cooling plates
- Sealings

The end plates are located on bottom and on top of the stack, the electrical current gets tapped on them. The bipolar plates form the flow channels for anode and cathode side. They are called bipolar plate because on one side of the plate the anode is placed and on the other side the cathode, that means on one side the plate has positive electrical charge and on the other it has negative electrical charge. In between the bipolar plates the MEA is located, it consists of the two electrodes and the electrolyte. On each side of the bipolar plates a sealing is inserted to prevent a mixing of anode, cathode and cooling fluids. Cooling plates have channels machined into them to

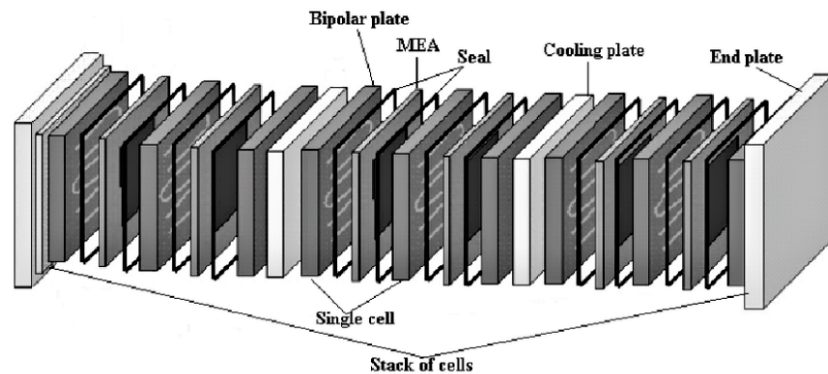


Figure 2.4: Schematic assembly of a fuel cell stack [17]

form the cooling channels. They are necessary to dissipate the heat from the stack via a cooling system.

2.1.2 Polymer Electrolyte Membrane FC

The PEM-FC is designed by utilizing a polymer electrolyte membrane. Usually, a perfluorinated sulphonic acid polymer is assembled in between two carbon-based electrodes. [11]

The electrodes are coated with platinum, which acts as a catalyst for the two half reactions presented in equation 2.1 & 2.2. [11]

The membrane, that acts as the electrolyte, is thin ($20\text{--}200\mu\text{m}$), transparent and flexible. The whole module of the two electrodes and the membrane is called membrane electrode assembly (MEA) and is approximately 1mm thick. [11]

In order to enable the membrane to conduct ions, it has to be hydrated by the product water. This restriction causes the setup to be operated at temperatures $< 90^{\circ}\text{C}$, so that the water isn't evaporated. Although typically the operating pressure is greater than 2 bar absolute, which increases the maximal operating temperatures due to evaporation. [11]

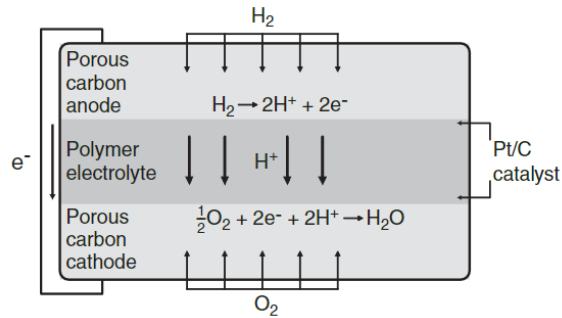


Figure 2.5: Schematic of H₂, O₂ PEM fuel cell[11]

PEM fuel cells are widely spread throughout the industry due to their highest power density of all fuel cell types. They also show a good resistance at on-off cycling and fast-startup. [11]

PEM FC Advantages [11]:

- High power density
- Good start-stop properties and dynamic load behaviour
- Low-temperature operation is beneficial at portable applications

PEM FC Disadvantages [11]:

- Use of expensive platinum
- Expensive MEA
- Active water management required
- Poor CO and S tolerance

2.2 Multiphysics CFD Solver

This chapter presents details to the most relevant basics of CFD. Four sub-chapters describe the Navier-Stokes equations (2.2.1), electromagnetic treatment (2.2.2), FC-Balancing equations (2.2.3) and finally the FC-Degradation modelling concepts (2.2.4)

2.2.1 Navier-Stokes equations

The Navier-Stokes-Equations model the interaction due to thermal and flow effects of fluids (gases & liquids). The equations describe how a velocity and pressure field evolves over time due to various forces and powers that act on the fluid.

The set of continuity equation 2.4, momentum equation 2.5 and energy equation 2.6 is called Navier-Stokes equations. They are obtained by applying a balance of mass, balance of momentum and balance of energy onto a infinitesimal volume element. The following equations are solved to obtain the pressure p , velocity \vec{v} , total internal energy e and temperature T . [16]

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{v}) = 0 [16] \quad (2.4)$$

$$\frac{\partial(\rho \vec{v})}{\partial t} + \nabla(\rho \vec{v} \vec{v}) = -\nabla p + \rho \vec{f} + \nabla \tau [16] \quad (2.5)$$

$$\frac{\partial e}{\partial t} + \nabla(\vec{v} e) = -\nabla \vec{v} p + \frac{\partial Q}{\partial t} + \rho \vec{f} \vec{v} - \nabla \vec{q} + \nabla(\tau \vec{v}) [16] \quad (2.6)$$

$$\vec{q} = -k \nabla T [16] \quad (2.7)$$

With \vec{f} describing volume sources (e.g., gravity), τ shear stresses, \dot{Q} heat sources and \vec{q} heat conduction. On the left-hand side, the first term of

each equation describes the local change within the infinitesimal volume element and the second term states for the convective portion of the variable exchange due to transport across the surface of the infinitesimal volume element. The right-hand side of the continuity equation is equal to zero, this means that mass is a conserved quantity. In the case of momentum equation, the right-hand side is equal to the sum of applied forces onto the infinitesimal volume and at the energy equation the right-hand side is equal to the power of the surface forces, heat conduction and heat sources. [5]

This results in a system of three partial differential equations for five unknowns. To close the system of equations two other relations are necessary. They are obtained by the thermodynamic state equations. By assuming ideal gas with constant specific heat (perfect gas) and a ratio of specific heats γ the following equations are described: [16]

$$e = \frac{p}{\gamma - 1} + \rho \frac{\vec{v}^2}{2} [16] \quad (2.8)$$

$$T = \frac{(\gamma - 1)}{R} \left(\frac{e}{\rho} - \frac{\vec{v}^2}{2} \right) [16] \quad (2.9)$$

The shear stresses are related to the flow field in combination with the viscosity μ . This relation is known assuming a Newtonian fluid. A Newtonian fluid is characterised by a linear dependency between deformation and resulting shear stress and is valid for most fluids.

$$\tau = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix} [16] \quad (2.10)$$

$$\tau_{i,j} = \mu \left[\frac{u_i}{x_j} + \frac{u_j}{x_i} + \delta_{i,j} (\nabla \cdot \vec{v}) \right] \quad \delta_{i,j} = 1 \text{ for } i = j \text{ else } 0 [16] \quad (2.11)$$

2.2.2 Electromagnetic

Electromagnetic effects are modelled by solving an electric scalar potential $\phi(\vec{x}, t)$ and a magnetic vector potential $\vec{A}(\vec{x}, t)$. To model the electric scalar potential the starting point is the charge conservation equation: [3]

$$\nabla \cdot \vec{i} = \dot{R} \quad [10] \quad (2.12)$$

\vec{i} is the current density and \dot{R} represents a current source. With the field equations of electromagnetic (Maxwell equations), this formulation can be expressed by the two variables solved in the CFD software:

$$-\nabla \cdot [\sigma \alpha \nabla \phi] = \dot{R} + \nabla \cdot \left[\sigma \alpha \left(\frac{\partial \vec{A}}{\partial t} - \vec{v} \times (\nabla \times \vec{A}) \right) \right] \quad [3] \quad (2.13)$$

σ describes the electrical conductivity tensor, α the phase volume fraction and \vec{v} the velocity of the material within the magnetic field. [3]

The following equation describes the dependence of the magnetic field \vec{H} with current, magnetic volume \vec{M}_V and surface sources \vec{M}_S , it is called amperes circuital law which is one of Maxwells equations.

$$\nabla \times \vec{H} = \vec{i} + \vec{M}_V + \nabla \times \vec{M}_S \quad [3] \quad (2.14)$$

Applying the definition of the magnetic vector potential, the relation between magnetic flux and magnetic field and the ohmic law gives: [3]

$$\sigma \alpha \left[\frac{\partial \vec{A}}{\partial t} - \vec{v} \times (\nabla \times \vec{A}) \right] + \nabla \times \frac{\nabla \times \vec{A}}{\mu} = -\sigma \alpha \nabla \phi + \vec{M}_V + \nabla \times \vec{M}_S \quad [3] \quad (2.15)$$

Equation 2.13 and 2.15 is a set of four partial differential equations that can be solved to determine $\phi(\vec{x}, t)$ and $\vec{A}(\vec{x}, t)$.

2.2.3 FC-Balancing Equations

No.	Solution Quantity	Balance Equation
(1)	p	$\frac{\partial(\alpha_g \rho_g + \alpha_l \rho_l)}{\partial t} + \nabla (\alpha_g \rho_g \vec{v}_g + \alpha_l \rho_l \vec{v}_l) =$
(2)	\vec{v}_g	$\frac{\partial(\alpha_g \rho_g \vec{v}_g)}{\partial t} + \nabla [\alpha_g \rho_g \vec{v}_g \vec{v}_g - \alpha_g \mu_g (\nabla \vec{v}_g + \nabla \vec{v}_g^T)] + \alpha_g \nabla p =$
(3)	\vec{v}_l	$\frac{\partial(\alpha_l \rho_l \vec{v}_l)}{\partial t} + \nabla [\alpha_l \rho_l \vec{v}_l \vec{v}_l - \alpha_l \mu_l (\nabla \vec{v}_l + \nabla \vec{v}_l^T)] + \alpha_l \nabla p =$
(4)	α_g	$\frac{\partial(\alpha_g \rho_g)}{\partial t} + \nabla (\alpha_g \rho_g \vec{v}_g) =$
(5)	α_l	$\frac{\partial(\alpha_l \rho_l)}{\partial t} + \nabla (\alpha_l \rho_l \vec{v}_l) =$
(6)	$y_{k,g}$	$\frac{\partial(\alpha_g \rho_g y_{k,g})}{\partial t} + \nabla (\alpha_g \rho_g \vec{v}_g y_{k,g} + \alpha_g \vec{j}_{k,g}) =$
(7)	C_w	$\frac{\partial}{\partial t} (\alpha_{ion} \frac{\rho_{ion}}{EW} C_w) + \nabla \vec{n}_w =$
(8)	$x_{k,w}$	$\frac{\partial}{\partial t} (\alpha_{ion} \frac{\rho_{ion}}{EW} C_w x_{k,w}) + \nabla (\vec{n}_w x_{k,w} - \alpha_{ion} \frac{\rho_{ion}}{EW} C_w D_{k,w} \nabla x_{k,w}) =$
(9)	$c_{k,agg}$	$\frac{\partial(\alpha_{ion,agg} c_{k,agg})}{\partial t} - \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 D_{k,w} \alpha_{ion,agg}^{1.5} \frac{\partial c_{k,agg}}{\partial r} \right) =$
(10)	ϕ_{ele}	$\nabla (-\alpha_{ele} \sigma_{ele} \nabla \phi_{ele}) =$
(11)	ϕ_{ion}	$\nabla (-\alpha_{ion} \sigma_{ion} \nabla \phi_{ion}) =$
(12)	T_g	$\frac{\partial(\alpha_g \rho_g H_g)}{\partial t} + \nabla (\alpha_g \rho_g \vec{v}_g H_g - \alpha_g \kappa_g \nabla T_g + \alpha_g \sum_k \vec{j}_{k,g} h_{k,g})$ $-\alpha_g \frac{\partial p}{\partial t} - \nabla [\alpha_g \mu_g (\nabla \vec{v}_g + \nabla \vec{v}_g^T) \cdot \vec{v}_g] =$
(13)	T_l	$\frac{\partial(\alpha_l \rho_l H_l)}{\partial t} + \nabla (\alpha_l \rho_l \vec{v}_l H_l - \alpha_l \kappa_l \nabla T_l) - \alpha_l \frac{\partial p}{\partial t}$ $-\nabla [\alpha_l \mu_l (\nabla \vec{v}_l + \nabla \vec{v}_l^T) \cdot \vec{v}_l] =$
(14)	T_s	$\frac{\partial(\alpha_s \rho_s H_s)}{\partial t} - \nabla (\alpha_s \kappa_s \nabla T_s) =$

Table 2.2: Balancing equations - fuel cell module [3]

No.	Solution Quantity	Sources
(1)	p	$= 0_{C,G} - S_{P,N,gd} - S_{P,N,ld}$
(2)	\vec{v}_g	$= -\vec{F}_{C,gl} + \vec{F}_{C,pc} + \vec{F}_{G,P,N,pc} + \vec{F}_{G,P,N,gs}$
(3)	\vec{v}_l	$= \vec{F}_{C,gl} - \vec{F}_{C,pc} - \vec{F}_{G,P,N,pc} + \vec{F}_{G,P,N,ls} + \vec{F}_{G,P,N,cap}$
(4)	α_g	$= S_{C,G,lg} - S_{P,N,gd} + S_{P,N,lg}$
(5)	α_l	$= -S_{C,G,lg} - S_{P,N,gd} - S_{P,N,lg}$
(6)	$y_{k,g}$	$= S_{C,G,lg} - S_{P,N,gd,k} + S_{P,N,lg}$
(7)	C_w	$= 0_M + \frac{S_{P,N,gd,w} + S_{P,N,ld}}{M_{H_2O}} + S_{P,ec,w}$
(8)	$x_{k,w}$	$= 0_M + \frac{S_{P,N,gd,k}}{M_k} - S_{N,par,k}$
(9)	$c_{k,agg}$	$= \frac{S_{P,ec,O_2}}{M_{O_2}} - \frac{S_{N,ec,H_2}}{M_{H_2}}$
(10)	ϕ_{ele}	$= 0_{G,B} - J_{P,ec} - J_{N,ec} + J_{N,ec,par}$
(11)	ϕ_{ion}	$= 0_M + J_{P,ec} + J_{N,ec} - J_{N,ec,par}$
(12)	T_g	$= Q_{C,lg} + Q_{C,pc} - Q_{G,P,N,gs} + Q_{G,P,N,lg} + Q_{G,P,N,pc}$
(13)	T_l	$= -Q_{C,lg} - Q_{C,pc} - Q_{G,P,N,ls} - Q_{G,P,N,lg} - Q_{G,P,N,pc}$
(14)	T_s	$= Q_{G,P,N,ls} + Q_{G,P,N,gs} + Q_{G,P,M,N,B,ohm} + Q_{P,N,ec}$

Table 2.3: Sources for balancing equations in table 2.2 [3]

Based on theory provided in chapters 2.2.1 and 2.2.2, a set of balancing equations is shown in table 2.2&2.3 to describe physics of galvanic cells. Table 2.2 states for the left hand side and table 2.3 states for the right hand side of the balancing equations. In table 2.3 indices show in which domain the equation is solved, only if there is at least on right hand side term the equation is solved in the specific domain.

Indices:

- C ... Gas Channel
- G ... Gas diffusion layer
- P ... Positive catalyst layer
- M ... Membrane
- N ... Negative catalyst layer
- B ... Bipolar plate

In Table 2.2 equation (1) is the continuity equation (conservation of mass), (2)&(3) are momentum equations for gaseous and liquid phase, (4)&(5) take care of the mass exchange between gaseous and liquid phase, (6) is a scalar transport equation modelling species transport for each species k , (7) describes the water transport equation of the dissolved gas phase, (8) describes the other k dissolved gas species equations (all species besides water), (9) models the diffusion of reactants into agglomerates (of platinum) where the chemical reactions take place according to local reactant concentration, (10)&(11) calculate the local potentials respectively for ionic and electric potential and finally (12)&(13)&(14) are energy equations for each phase. [3]

2.2.4 FC-Degradation Modelling

Degradation models are well documented in the AVL-FireM[®] manual on which this overview is mainly based on. [3] The following degradation mechanisms are available:

- Ionomer Degradation [19]
- Carbon corrosion, carbon oxidation, platinum oxidation [7] [13]
- Platinum dissolution and redeposition [7]
- Particle detachment and agglomeration [7]

The **ionomer degradation** model takes H_2O_2 into account, that is formed in the catalyst layers and diffuses into the membrane. In the membrane it reacts with ferrous ions to OH radicals. This process is called the Fenton's reaction and leads to: [3]

- Loss of sulphonic acid groups in the polymer within the membrane, which decreases the membrane ionic conductivity
- Thinning of membrane as well as the catalyst layers
- HF (hydrogen fluoride) and CO_2 production

All other degradation mechanisms focus on the cathode catalyst layer. The idea of these models is to cluster the platinum into particle size classes with constant geometric properties. The models describe processes that cause the smaller particles to shift into the greater particle size classes. The result is a decreased electrochemical surface area (ECSA). [7]

The **corrosion** and **oxidation** mechanisms produce CO_2 and therefore decreases the carbon mass in the catalyst layer. This effect is considered in each particle class separately and causes platinum particle detachment. [3]

Platinum dissolution and redeposition means that platinum ions, which lack a oxidation cover, are able to diffuse into the membrane. Redeposition takes place when a large concentration of platinum ions is present, that prevents the ions to diffuse into the membrane and it becomes more likely that they redeposit onto large platinum particles. This process is called "Ostwald ripening" and cause a decrease of ECSA and an increase of the mean Pt-particle size. [3]

Particle detachment and agglomeration is considered by enabling atoms to detach and agglomerate onto existing particles. Atoms become loose due to carbon corrosion. This process will alter the particle size distribution by increasing the mean particle size and decreasing the ECSA. The models do not constitute the way that particles move, every atom is agglomerated instantly after detaching. The model assumes that all particle sizes are equal likely to receive a detached atom and that detached platinum will always agglomerate onto an existing particle group and not form new particle groups. [3]

Simulations are performed transient with a sub-time stepping. Local stressors and sources are recorded during a fine time resolution of a cyclic repeating voltage or current cycle (e.g. 6s). Afterwards a large time step (e.g. 6000s) is applied in which the recorded stressors are applied in a sub-time step. Local stressors are local voltage, temperature, humidity and oxygen concentration. Feedback to performance as well as an update of the cycle can be performed by the frequency of the large time step. This method establishes the simulation of accelerated stress tests, which are carried out on test-bed environments as a state-of-the-art benchmarking of design in terms of degradation. [16]

The following material properties are adapted, due to degradation, according to AVL-FireM[®] manual. [3] The adaption of the material properties ensures that the model keeps track on the performance drop due to degradation.

- Decreasing solid material in the CL and membrane thicknesses
- Platinum volume fractions in the CL are calculated according to mass balances
- Agglomerate radius and ionomer film thickness in the CL
- Platinum particle size distribution in the CL
- Ionic conductivity and equivalent weight in the membrane

Ionomer degradation

No.	Reaction
(1)	$O_2 + 2H^+ + 2e^- \longrightarrow H_2O_2 \quad \phi_{eq,id,1,0} = 0.695VRHE$
(2)	$H_2O_2 + 2H^+ + 2e^- \longrightarrow 2H_2O \quad \phi_{eq,id,2,0} = 1.76VRHE$
(3)	$Fe^{2+} + H_2O_2 + H^+ \longrightarrow Fe^{3+} + OH + H_2O$
(4)	$R_fSO_3^- + OH \longrightarrow R_{f\alpha}O + HO CF_2CF_2SO_3^-$
(5)	$R_{f\alpha}O + 3OH \longrightarrow R_{f\beta}O + CF_3 + 3HF + 2CO_2$
(6)	$R_{f\beta}O + 2H_2O + OH \longrightarrow 2-(CF_2)_{n_0}COOH + 3HF$
(7)	$-(CF_2)_jCOOH + 2OH \longrightarrow CO_2 + 2HF + -(CF_2)_{j-1}COOH$

Table 2.4: Ionomer degradation reactions [3] [19]

Table 2.4 shows the ionomer degradation reactions according to reference [19]. Reaction (1) describes the production of hydrogen peroxide, due to the presence of oxygen and hydrogen ions, (2) shows the reaction of hydrogen peroxide with hydrogen ions to water. Reaction (3) (4) (5) determine the side chain degradation of the perfluorosulphonic acid groups of the membrane. [19] R_f is a placeholder for the basis molecule from which the degradation effects cleavage the side chains. α and β signalize different stages of the degraded molecule. Reaction (6) is a main chain scission and (7) shows the unzipping reaction of the products from (6). Details are well documented in reference [19].

$$j_{id,1} = -j_{0,id,1} \frac{c_{O_2,g}}{c_{ref,g}} \left(\frac{c_{H^+}}{c_{ref,H^+}} \right)^2 e^{\left(-\frac{2\beta_{id,1}F}{RT_s} \eta_{act,id,1} \right)} [3] \quad (2.16)$$

$$j_{id,2} = j_{0,id,2} \frac{c_{H_2O_2,g}}{c_{ref,g}} \left[e^{\left(\frac{2(1-\beta_{id,2})F}{RT_s} \eta_{act,id,2} \right)} - e^{\left(-\frac{2\beta_{id,2}F}{RT_s} \eta_{act,id,2} \right)} \right] [3] \quad (2.17)$$

$$\dot{r}_{id,3} = k_{f,id,3} c_{H_2O_2,w} c_{Fe^{2+},w} [3] \quad (2.18)$$

$$\dot{r}_{id,4} = k_{f,id,4} c_{R_fSO_3,w} c_{OH,w} [3] \quad (2.19)$$

$$\dot{r}_{id,5} = k_{f,id,5} c_{R_f\alpha O,w} c_{OH,w} [3] \quad (2.20)$$

$$\dot{r}_{id,6} = k_{f,id,6} c_{R_f\beta O,w} c_{OH,w} [3] \quad (2.21)$$

$$\dot{r}_{id,7} = k_{f,id,7} c_{(CF_2)_nCOOH,w} c_{OH,w} [3] \quad (2.22)$$

Equations 2.16 and 2.17 determines the reaction current density according to the Butler-Volmer formulation. $j_{0,id,i}$ is the reference exchange current density, $c_{j,k}$ the molar concentration, $\eta_{act,id,i}$ the activation overpotential and $\beta_{id,i}$ the charge transfer coefficient. Reaction rates for 2.18 - 2.22 are determined by the molar concentrations c and the reaction rate constants $k_{f,id,i}$. [3] The activation overpotential $\eta_{act,id,i}$ is calculated by the ionic, electric and equilibrium potential. [3] The equilibrium potential is given in table 2.4 and can be calculated according to equation 2.36.

$$\eta_{act,id,j} = \phi_{ele} - \phi_{ion} - \phi_{eq,id,j} \text{ for } j = 1, 2 \quad (2.23)$$

The following equations show the calculation of the ferrous/hydrogen ion concentrations. Fe^{2+} ions are assumed to be present within the membrane as a constant contamination. [19] [3]

$$c_{H^+} = \frac{\rho_{ion} + 32.4 C_w}{(1 + 0.0648 C_w) EW} \quad (2.24)$$

$$c_{Fe^{2+},w} = \frac{y_{Fe^{2+},w} \rho_{ion}}{M_{Fe^{2+}}} \quad (2.25)$$

C_w is the dissolved water content, EW the equivalent weight of the ionomer and $y_{Fe^{2+}}$ is the mass fraction of Fe^{2+} ions. Due to the reactions a set

of mass sources are applied to the dissolved gas species balance equations 2.2(8), they are mainly related to the reaction rates/current densities [3]. Reaction (2) causes a water source in the catalyst layers. Furthermore balance equations of the ionomer species are solved for $k = OH, R_fSO_3, R_f\alpha O, R_f\beta O, (CF_2)_nCOOH, MC, CF_2$:

$$\frac{\partial (\alpha_{ion} c_{k,ion})}{\partial t} = S_{id,k} \alpha_{ion} \quad (2.26)$$

The current J_{id} produced by the electrochemical reactions and is added to the ionic charge conservation equation 2.2(11) and subtracted from the electronic charge conservation equation 2.2(10). Heat is also produced and considered via the solid enthalpy equation. 2.2(14)

$$J_{id} = (j_{id,1} + j_{id,2}) a_{Pt} \quad (2.27)$$

$$Q_{id} = (j_{id,1} \eta_{act,id,1} + j_{id,2} \eta_{act,id,2}) a_{Pt} \quad (2.28)$$

Carbon corrosion, carbon oxidation, platinum oxidation modelling

No.	Reaction	
(1)	$C^* + H_2O \rightleftharpoons C-OH + H^+ + e^-$	$\phi_{eq,cc,1,0} = 0.29VRHE$
(2)	$C-OH \rightleftharpoons C=O + H^+ + e^-$	$\phi_{eq,cc,2,0} = 0.80VRHE$
(3)	$C^* - C - OH + H_2O \longrightarrow$ $C^* + CO_2 + 3H^+ + 3e^-$	$\phi_{eq,cc,3,0} = 0.96VRHE$
(4)	$Pt^* + H_2O \rightleftharpoons Pt(OH)_{ads} + H^+ + e^-$	$\phi_{eq,cc,4,0} = 0.79VRHE$
(5)	$Pt(OH)_{ads} \rightleftharpoons PtO_{ads} + H^+ + e^-$	$\phi_{eq,cc,5,0} = 0.80VRHE$
(6)	$Pt(OH)_{ads} + C^* - C - OH \longrightarrow$ $C^* + CO_2 + Pt^* + 2H^+ + 2e^-$	$\phi_{eq,cc,6,0} = 0.62VRHE$

Table 2.5: Carbon corrosion reactions [3] [7]

In table 2.5, reaction (3) & (6) describe carbon corrosion, (1) & (2) carbon oxidation and (4) & (5) platinum oxidation. Atoms marked with a star (*) indicate suitable defect sites on the structure. These defects are necessary for the reactions. The equilibrium potentials are given in reference to a reversible hydrogen electrode (RHE). [7]

$$j_{cc,1,i} = j_{0,cc,1} \left[\varphi_{H_2O} f_{C,i} e^{\left(\frac{\beta_{cc,1} F}{R_m T_s} \eta_{act,cc,1,i} \right)} - f_{COH,i} e^{-\left(\frac{(1-\beta_{cc,1}) F}{R_m T_s} \eta_{act,cc,1,i} \right)} \right] [3] \quad (2.29)$$

$$j_{cc,2,i} = j_{0,cc,2} \left[f_{COH,i} e^{\left(\frac{\beta_{cc,2} F}{R_m T_s} \eta_{act,cc,2,i} \right)} - f_{CO,i} e^{-\left(\frac{(1-\beta_{cc,2}) F}{R_m T_s} \eta_{act,cc,2,i} \right)} \right] [3] \quad (2.30)$$

$$j_{cc,3,i} = j_{0,cc,3} \varphi_{H_2O} f_{COH,i} e^{\left(\frac{3\beta_{cc,3} F}{R_m T_s} \eta_{act,cc,3,i} \right)} [3] \quad (2.31)$$

2 Theoretical Part

$$j_{cc,4,i} = j_{0,cc,4} \left[\varphi_{H_2O} f_{Pt,i} e^{\left(\frac{\beta_{cc,4} F}{R_m T_s} (\eta_{act,cc,4,i} - r_{OX} (f_{PtOH,i} + f_{PtO,i})) \right)} \right] [3] \quad (2.32)$$

$$j_{cc,5,i} = j_{0,cc,5} \left[f_{PtOH,i} e^{\left(\frac{\beta_{cc,5} F}{R_m T_s} (\eta_{act,cc,5,i} - r_{OX} (f_{PtOH,i} + f_{PtO,i})) \right)} \right] [3] \quad (2.33)$$

$$j_{cc,6,i} = j_{0,cc,6} f_{PtOH,i} f_{COH,i} e^{\left(\frac{2\beta_{cc,6} F}{R_m T_s} \eta_{act,cc,6,i} \right)} [3] \quad (2.34)$$

Equation 2.29 - 2.34 show reaction current densities corresponding to the reactions in table 2.5. Reaction current densities are calculated according to the Butler-Volmer-Equation. $j_{0,cc,j}$ is the exchange current density, it is related to the reaction rates and therefore how fast a reaction takes place. Reaction current densities are modelled per surface area carbon or platinum for each particle size i . φ_{H_2O} describes relative humidity, $f_{j,i}$ the surface fraction, $\beta_{cc,j}$ the charge transfer coefficient and $\eta_{act,cc,j}$ the activation overpotential. $\eta_{act,cc,j}$ is calculated by the ionic, electric and equilibrium potential. [3]

$$\eta_{act,cc,j,i} = \phi_{ele} - \phi_{ion} - \phi_{eq,cc,j,i} [3] \quad (2.35)$$

The equilibrium potential $\phi_{eq,cc,j}$ for reaction (1), (2) & (3) is equal to $\phi_{eq,cc,j,0}$. This implicates that the carbon atoms form a flat support structure and therefore do not need another modification term. This is a good assumption due to the relative to platinum, great amount of carbon. $\phi_{eq,cc,j,0}$ is calculated according to equation 2.36.

$$ze\phi_{eq,cc,j,0} = \Delta_R H - T\Delta_R S = \Delta_R G_m [2] \quad (2.36)$$

$\Delta_R G_m$ is the difference of the molar Gibbs free energy between reactants and products of a chemical reaction. z is the number of electrons transferred at the reaction. Due to the particle size class approach at the platinum related reactions, equilibrium potentials are shifted according to the radius of the

particle classes and the surface tension γ . This modification is called Kelvin shift. [3]

$$\phi_{eq,cc,4,i} = \phi_{eq,cc,4,0} - \frac{1}{2Fr_{Pt,i}} \left(\frac{\gamma_{Pt} M_{Pt}}{\rho_{Pt}} - \frac{\gamma_{PtOH} M_{PtOH}}{\rho_{PtOH}} \right) [3] \quad (2.37)$$

$$\phi_{eq,cc,5,i} = \phi_{eq,cc,5,0} - \frac{1}{2Fr_{Pt,i}} \left(\frac{\gamma_{PtOH} M_{PtOH}}{\rho_{PtOH}} - \frac{\gamma_{PtO} M_{PtO}}{\rho_{PtO}} \right) [3] \quad (2.38)$$

$$\phi_{eq,cc,6,i} = \phi_{eq,cc,6,0} - \frac{1}{2Fr_{Pt,i}} \frac{\gamma_{PtOH} M_{PtOH}}{\rho_{PtOH}} [3] \quad (2.39)$$

The following conservation equations describe the amount of surface fractions $f_{j,i}$ respectively for the solid reactants and products j present in the oxidation/corrosion reactions from table 2.5, they are determined for each particle size class i : [3]

$$\psi_{dec} \gamma_C \frac{df_{COH,i}}{dt} = j_{cc,1,i} - j_{cc,2,i} - j_{cc,3,i} - j_{cc,6,i} [3] \quad (2.40)$$

$$\psi_{dec} \gamma_C \frac{df_{CO,i}}{dt} = j_{cc,2,i} [3] \quad (2.41)$$

$$\psi_{dec} \gamma_C \frac{df_{PtOH,i}}{dt} = j_{cc,4,i} - j_{cc,5,i} - \frac{d_{cc}}{r_{Pt,i}} j_{cc,6,i} [3] \quad (2.42)$$

$$\psi_{dec} \gamma_{Pt} \frac{df_{PtO,i}}{dt} = j_{cc,5,i} [3] \quad (2.43)$$

$$f_{C,i} = 1 - f_{COH,i} - f_{CO,i} [3] \quad (2.44)$$

2 Theoretical Part

$$f_{Pt,i} = 1 - f_{PtOH,i} - f_{PtO,i} [3] \quad (2.45)$$

γ_{Pt} and γ_C are surface densities of the suitable defect sites which are necessary to enable the reactions. d_{cc} is the carbon corrosion depth. A model parameter is implemented to tune the corrosion/oxidation effects globally, if it is >1 the reactions are decelerated. The produced CO_2 is added, via a source term, in the conservation equation of the dissolved molar species 2.2(8) in the cathode catalyst layer. [7]

$$S_{cc,CO2} = \frac{1}{\psi_{dec} F} \sum_{i=1}^n (a_{C,i} j_{cc,3,i} + 2\pi r_{Pt,i} d_{cc} n_{Pt,i} j_{cc,6,i}) [3] \quad (2.46)$$

$a_{C,i}$ is the carbon surface density, respectively for each particle size class i.

$$a_{C,i} = a_{C,bol} f_{C,i} \frac{n_{Pt,i}}{n_{Pt,tot}} [3] \quad (2.47)$$

The weighting according to the platinum number density $n_{Pt,i}$ represents the assumption of evenly distributed carbon between the platinum particles. $a_{C,bol}$ is the carbon surface density at begin of life. [3]

$$a_{C,bol} = A_{C,bol} \rho_C \left(1 - \epsilon_{bol} - \alpha_{ion,bol} - \frac{m_{Pt,bol}}{\rho_{Pt} \nabla_{cl,bol}} \right) [3] \quad (2.48)$$

The reactions produce heat Q_{cc} and additional current J_{cc} in the catalyst layer, the current is added to the ionic charge conservation equation 2.2(11) and subtracted from the electronic charge conservation equation 2.2(10). The heat is added to the solid enthalpy equation 2.2(14). Sources are applied for each reaction j and particle size class i. [3]

$$J_{cc} = \frac{1}{\psi_{dec}} \sum_{i=1}^n \sum_{j=1}^6 j_{cc,j,i} a_{j,i} [3] \quad (2.49)$$

$$Q_{cc} = \frac{1}{\psi_{dec}} \sum_{i=1}^n \sum_{j=1}^6 j_{cc,j,i} a_{j,i} \eta_{act,cc,j,i} [3] \quad (2.50)$$

The specific surface densities of the carbon-based reactions are calculated as described in 2.48. At the platinum-based reactions it is calculated by the surface of platinum assuming spheres. [3]

$$a_{1,i} = a_{2,i} = a_{3,i} = a_{C,i} [3] \quad (2.51)$$

$$a_{4,i} = a_{5,i} = a_{Pt,i} = 4\pi r_{Pt,i}^2 n_{Pt,i} [3] \quad (2.52)$$

To determine the specific surface density of the carbon corrosion reaction (6), it is assumed that the reaction only takes place within a ring disk area around the platinum sphere on the carbon surface plane. The width of this ring disk is the carbon corrosion depth d_{cc} . [7]

$$a_{6,i} = 2\pi r_{Pt,i} n_{Pt,i} d_{cc} [3] \quad (2.53)$$

Platinum dissolution and redeposition

No.	Reaction
(1)	$\text{Pt} \rightleftharpoons \text{Pt}^{2+} + \text{e}^- \quad \phi_{eq,pd,0} = 1.155 \text{VRHE}$

Table 2.6: Platinum dissolution reaction [3] [7]

$$j_{pd,i} = j_{0,pd} f_{Pt,i} \left[e^{\left(\frac{2\beta_{pd}F}{RT_s} \eta_{act,pd,i} \right)} - \frac{c_{Pt2+}}{c_{Pt2+,ref}} e^{-\left(\frac{2(1-\beta_{pd})F}{R_m T_s} \eta_{act,pd,i} \right)} \right] [3] \quad (2.54)$$

Equation 2.54 show reaction current densities corresponding to the platinum dissolution reaction in table 2.6. Reaction current densities are calculated according to the Butler-Volmer-Equation. $j_{0,pd}$ is the exchange current density. Reaction current densities are modelled per surface area platinum for each particle size i . $f_{Pt,i}$ describes the platinum surface fraction, β_{pd} the charge transfer coefficient, $c_{Pt2+,ref}$ the reference platinum ion concentration, c_{Pt2+} the platinum ion concentration and $\eta_{act,pd,i}$ the activation overpotential. $\eta_{act,pd,i}$ is calculated by the ionic-, electric- and equilibrium potential. [3]

$$\eta_{act,pd,i} = \phi_{ele} - \phi_{ion} - \phi_{eq,pd,i} [3] \quad (2.55)$$

$$\phi_{eq,pd,i} = \phi_{eq,pd,0} - \frac{1}{2Fr_{Pt,i}} \frac{\gamma_{Pt} M_{Pt}}{\rho_{Pt}} [3] \quad (2.56)$$

The platinum dissolution model assumes the reference platinum ion concentration $c_{Pt2+,ref}$ with 1000 mol/m³, a low value will produce less dissolution and redeposition becomes more likely. To determine the local platinum ion concentration c_{Pt2+} , the following platinum ion mass balance is solved within the catalysts layers and the membrane:

$$\frac{\partial(\alpha_{ion}c_{Pt2+})}{\partial t} - \nabla(\alpha_{ion}D_{Pt2+,w}\nabla c_{Pt2+}) = S_{pd,Pt2+} + S_{red,Pt2+} [3] \quad (2.57)$$

α_{ion} is the ionic phase volume fraction, $D_{Pt2+,w}$ the diffusion coefficient of platinum ions within dissolved water, $S_{pd,Pt2+}$ the volumetric platinum ion source due to dissolution.

$$S_{pd,Pt2+} = -\frac{4\pi}{3} \frac{\rho_{Pt}}{M_{Pt}} \sum_{i=1}^n (r_{Pt,i}^3 \dot{n}_{Pt,pd,i}) [3] \quad (2.58)$$

$\dot{n}_{Pt,pd,i}$ is the rate of change of the particle number density due to Pt dissolution. The following formulation is valid for an equidistant distribution with a uniform class width of Δr_{Pt} . The growth or shrinkage of a specific particle size class leads to a reduction of the number of particles within this class, but there is also the possibility that another growing/shrinking particle size class contributes and therefore increases the number of particles within this class. Positive reaction current densities mean that the products (Pt^{2+} ions) increase and the reactants (platinum) decrease and therefore the particles within that size class shrink. Notice that shrinking particles from the smallest class 1 can vanish but growing particles from the biggest class are kept within the biggest class. Therefore the following set of equations 2.59-2.61 are not conservation equations. [7]

$$\dot{n}_{Pt,pd,1} = \frac{1}{F\Delta r_{Pt}} \frac{M_{Pt}}{\rho_{Pt}} [-n_{Pt,1}|j_{pd,1}| + n_{Pt,2}j_{pd,2}H(j_{pd,2})] [3] \quad (2.59)$$

$$\begin{aligned} \dot{n}_{Pt,pd,i} = \frac{1}{2F\Delta r_{Pt}} \frac{M_{Pt}}{\rho_{Pt}} [n_{Pt,i}(j_{pd,i+1} - j_{pd,i-1}) + j_{pd,i}(n_{Pt,i+1} - n_{Pt,i-1})] \\ \text{for } 2 \leq i \leq n-1 [3] \end{aligned} \quad (2.60)$$

2 Theoretical Part

$$\dot{n}_{Pt,pd,n} = -\frac{1}{F\Delta r_{Pt}} \frac{M_{Pt}}{\rho_{Pt}} [n_{Pt,n-1} j_{pd,n-1} H(-j_{pd,n-1}) + n_{Pt,n} j_{pd,n} H(j_{pd,n})] [3] \quad (2.61)$$

In the limit of infinite number of particle size classes, the formulation transforms into the expression for a continuous distribution. [7]

The platinum dissolution reaction contributes heat Q_{pd} to the solid enthalpy equation 2.2(14) and current J_{pd} , which is subtracted from the electronic charge conservation equation 2.2(10) and added to the ionic charge conservation equation 2.2(11).

$$J_{pd} = \sum_{i=1}^n j_{pd,i} a_{Pt,i} [3] \quad (2.62)$$

$$Q_{pd} = \sum_{i=1}^n j_{pd,i} \eta_{act,pd,i} a_{Pt,i} [3] \quad (2.63)$$

Particle detachment and agglomeration

The change of platinum number density for each particle size class i is calculated from three contributors: detachment, attachment, and merging.

$$\dot{n}_{Pt,agg,i} = -\dot{n}_{Pt,det,i} - \dot{n}_{Pt,att,i} + \dot{n}_{Pt,mer,i} [3] \quad (2.64)$$

$$\dot{n}_{Pt,det,i} = k_{det} \frac{1}{F\psi_{dec}} \frac{n_{Pt,i}}{r_{Pt,i}} (j_{cc,3,i} + j_{cc,6,i}) [3] \quad (2.65)$$

Detachment is directly proportional to the amount of carbon corrosion via their current densities. k_{det} is a parameter to account for all effects that are not considered within the simple formulation e.g., defects within the carbon surface or the exact topology between platinum and its carbon support. Note that smaller particles are more likely to detach than larger ones. [7]

$$\dot{n}_{Pt,det,tot} = \sum_{i=1}^n \dot{n}_{Pt,det,i} [3] \quad (2.66)$$

$$\dot{n}_{Pt,att,i} = \dot{n}_{Pt,det,tot} \frac{n_{Pt,i}}{n_{Pt,tot}} [3] \quad (2.67)$$

The model assumes that detached particles merge with attached particles independent of their particle size. The rate of attachment to detached particles, due to merging, is proportional to the total detachment rate and weighted to the amount of platinum particles of size class i . [7]

$$\dot{n}_{Pt,mer,i} = \frac{r_{Pt,i} X_S}{N_i \Delta r_{Pt}} \sum_{\sqrt[3]{r_{Pt,j}^3 + r_{Pt,k}^3} \approx r_{Pt,i}} \frac{\dot{n}_{Pt,det,j} \dot{n}_{Pt,att,k}}{\dot{n}_{Pt,det,tot}} [3] \quad (2.68)$$

The formulation of the merging is based on a continuous particle size distribution [15] and reformulated to a discrete amount of particle size classes [7]. To consider all possible pairs of $r_{pt,j}$ and $r_{pt,k}$ contributing to the merged particles $r_{pt,i}$, the sum has to be corrected by the fraction of terms in the discrete sum (N_i) to the area of integration in the continuous case ($X_S \approx 1.766$). [7]

2.3 Machine Learning

The aim of this chapter is to present theory to the Machine Learning concepts that are utilized in this thesis. At first the Principal Component Analysis is described (2.3.1). Afterwards the Kernel Ridge Regression method is presented (2.3.2) and finally the basics of a Dense Neural Network are introduced 2.3.3.

2.3.1 Principal Component Analysis

Machine Learning models are often utilized to process data with large amounts of features (hundreds or even millions). Usually computational power is limited, that leads to the idea of reducing the dimensionality of the features. The smaller number of features and samples, the less the computational effort.

The main idea is to project data from a high dimensional space into a lower dimensional space, without discarding the vast amount of information that is available from data. The decision, on which lower dimensional space projections should be applied, is made by the concept of preserving the variance.

Let's look at the left part from figure 2.6, it shows a two dimensional dataset. PCA determines the axis (c1 in figure 2.6 - top right) on which the most variance is preserved. [4]

Depending on which is smaller, the amount of datapoints (N) or the dimensionality of the datapoints (M), the PCA algorithm calculates N or

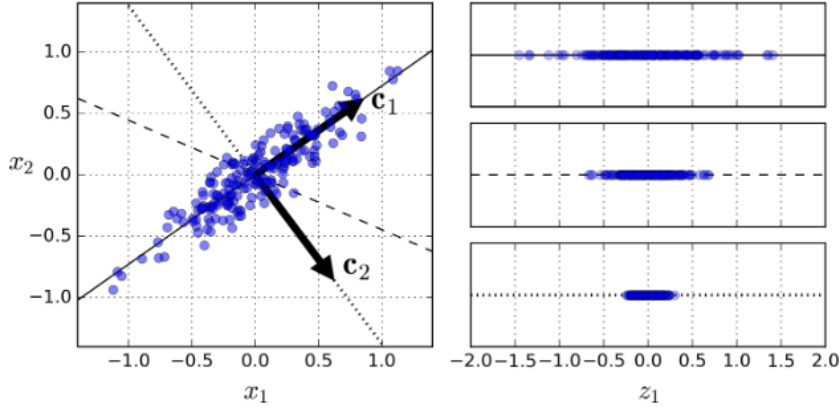


Figure 2.6: PCA subspace decision visualization [4]

M principal components (axis). The principal components are ordered according to their eigenvalues, the biggest eigenvalue correspond to the axis on which the most variance is preserved (most information is available). One has to decide how much information of the original dataset should be kept. This decision is made by comparing eigenvalues of each principal component and defining the amount of information that should be kept. The original data is projected onto the principal components to determine the reduced data. [4]

Let's assume N samples of M dimensional feature vectors that form the $M \times N$ dimensional data matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$:

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_N^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_N^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{(M)} & x_2^{(M)} & \dots & x_N^{(M)} \end{bmatrix} \quad [8] \quad (2.69)$$

The target of PCA is to find the transformation matrix ($\mathbf{W} \in \mathbb{R}^{M \times m}$) to calculate the low dimensional feature vector ($\mathbf{y}_i \in \mathbb{R}^m$) from the high dimensional feature vector ($\mathbf{x}_i \in \mathbb{R}^M$): [8]

$$\mathbf{y}_i = \mathbf{W}^T \mathbf{x}_i = \left[y^{(1)}, y^{(2)}, \dots, y^{(m)} \right] \quad \text{with } m \ll M [8] \quad (2.70)$$

The reconstruction from m to M dimensions ($\hat{\mathbf{x}}_i \in \mathbb{R}^M$) is calculated with the transformation matrix $\mathbf{Z} \in \mathbb{R}^{M \times m}$:

$$\hat{\mathbf{x}}_i = \mathbf{Z} \mathbf{y}_i [8] \quad (2.71)$$

We aim to design $\mathbf{W} \in \mathbb{R}^{M \times m}$ and $\mathbf{Z} \in \mathbb{R}^{M \times m}$ in a way such the quadratic error ($\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$) between the feature vector ($\mathbf{x}_i \in \mathbb{R}^M$) and the reconstruction from the lower dimensional space ($\hat{\mathbf{x}}_i \in \mathbb{R}^M$) is minimized [8]:

$$\operatorname{argmin} \sum_{n=1}^N \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 = \sum_{n=1}^N \|\mathbf{x}_i - \mathbf{Z} \mathbf{y}_i\|^2 = \sum_{n=1}^N \|\mathbf{x}_i - \mathbf{Z} \mathbf{W}^T \mathbf{x}_i\|^2 [8] \quad (2.72)$$

The optimization problem above can be solved by setting $\mathbf{Z} = \mathbf{W}$. [8]

One way of computing the principal components \mathbf{W} , is to utilize Singular Value Decomposition on the original data $\mathbf{X} \in \mathbb{R}^{M \times N}$. [8]

$$\mathbf{X} = \mathbf{V} \begin{bmatrix} \mathbf{D} & \mathbf{0} \end{bmatrix} \mathbf{U} \quad \text{if } N > M \qquad \mathbf{X} = \mathbf{V} \begin{bmatrix} \mathbf{D} \\ \mathbf{0} \end{bmatrix} \mathbf{U} \quad \text{if } M > N \quad (2.73)$$

Here $\mathbf{V} \in \mathbb{R}^{M \times M}$ (left eigenvectors) and $\mathbf{U} \in \mathbb{R}^{N \times N}$ (right eigenvectors) are the unitary matrices and $\mathbf{D} \in \mathbb{R}^{M \times M}$ or $\mathbb{R}^{N \times N}$ (singular values) is the diagonal matrix. The diagonal matrix \mathbf{D} is formed with the criteria $\min(M, N)$, so depending on M and N the diagonal matrix is formed by the smaller one.

In order to calculate the lower dimensional feature vector $\mathbf{y}_i \in \mathbb{R}^m$ from the higher dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^M$, utilizing SVD, Equation 2.70 becomes:

$$y_i = V_m^T x_i [8] \quad (2.74)$$

Let $y_i \in \mathbb{R}^m$ represent the reduced order representation calculated with a feature vector $x_i \in \mathbb{R}^M$ and $V_m \in \mathbb{R}^{M \times m}$ created with the first m columns of the unitary matrix $V \in \mathbb{R}^{M \times M}$. [8]

2.3.2 Kernel Ridge Regression

Kernel Ridge Regression is a kernelized version of Ridge Regression. Ridge Regression is a linear model that minimizes a squared error between a prediction and a ground truth value and introduces a quadratic penalty term in order to keep the model parameters (weights) small. Using such penalty terms to prevent model parameters from becoming too big is called regularization. [8] Small model parameters will decrease the risk of overfitting. [4]

The following objective function is minimized to determine the best model parameters vector $w \in \mathbb{R}^M$. [8]

$$J_{RR}(w) = \sum_{i=1}^N \epsilon_i^2 + \rho ||w||^2 [8] \quad (2.75)$$

$$\epsilon_i = w^T x_i - y_i [8] \quad (2.76)$$

J_{RR} is the cost function of Ridge Regression, $x_i \in \mathbb{R}^M$ the feature vector, $w \in \mathbb{R}^M$ the Ridge Regression weight vector (model parameters), ρ the regularization parameter, ϵ the error term and $y_i \in \mathbb{R}^1$ the label which should be predicted by the model after training. [8]

The training data matrix ($X \in \mathbb{R}^{M \times N}$) denotes a $M \times N$ matrix that consists of all feature vectors $x_i \in \mathbb{R}^M$. [8]

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_N] \quad [8] \quad (2.77)$$

The training data vector ($\mathbf{y} \in \mathbb{R}^N$) denotes a N dimensional vector that consists of all label values $y_i \in \mathbb{R}^1$. [8]

$$\mathbf{y} = [y_1, y_2 \dots y_N] \quad [8] \quad (2.78)$$

The matrix ($\mathbf{S} \in \mathbb{R}^{M \times M}$) denotes a $M \times M$ scatter matrix. [8]

$$\mathbf{S} = \mathbf{X}\mathbf{X}^T [8] \quad (2.79)$$

The solution to the optimization problem (minimizing the cost function Equation 2.75) in matrix form is the following, with \mathbf{I} denoting the identity matrix: [8]

$$\mathbf{w} = [\mathbf{X}\mathbf{X}^T + \rho\mathbf{I}]^{-1} \mathbf{X}\mathbf{y} = [\mathbf{S} + \rho\mathbf{I}]^{-1} \mathbf{X}\mathbf{y} [8] \quad (2.80)$$

Ridge Regression is a linear model. To make it predict non-linear behaviour, there are several approaches to do so. One simple way to make non-linear predictions is to add features with higher dimensions to the model. Instead of feeding a \mathbf{x}_i feature vector to the model, we can also add \mathbf{x}_i^2 or \mathbf{x}_i^3 to the model. e.g. by adding squared features \mathbf{x}_i^2 and rejecting the original features \mathbf{x}_i , the model will be able to picture quadratic correlations of the features to the labels, but will lack linear correlations.[4]

If the linear and quadratic features are fed to the model, the total amount of features is increased from M features to $2M$ features (or J features, so J is defining the total amount of features in the intrinsic space). The transformation from the original feature space to the intrinsic space is described by the representative vector $\phi(\mathbf{x}) \in \mathbb{R}^J$:

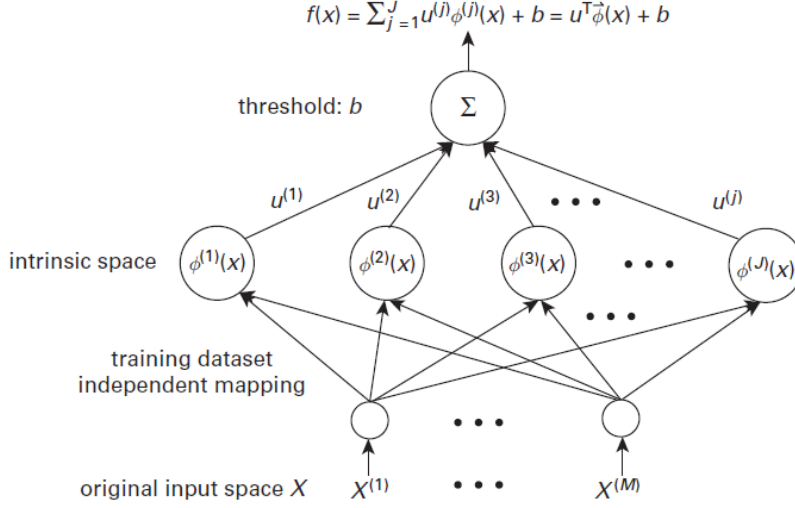


Figure 2.7: Transformation from original feature space with M features into intrinsic feature space with J features [8]

$$\boldsymbol{\phi}(x) = [\phi^{(1)}(x), \phi^{(2)}(x) \dots \phi^{(J)}(x)] \quad [8] \quad (2.81)$$

The resulting features from the transformation with the representative vector $\boldsymbol{\phi}(x) \in \mathbb{R}^J$ over all N samples are arranged in the matrix $\boldsymbol{\Phi}(x) \in \mathbb{R}^{J \times N}$

$$\boldsymbol{\Phi}(x) = [\boldsymbol{\phi}(x_1), \boldsymbol{\phi}(x_2) \dots \boldsymbol{\phi}(x_N)] \quad [8] \quad (2.82)$$

Figure 2.7 shows a two layer network that illustrates how original features can be extended to allow the linear model to make predictions in a non-linear manner. $f(x)$ describes the prediction function, $\mathbf{u} \in \mathbb{R}^J$ are the transformed model parameter vector and b is the bias term. [8]

Another way to make non-linear predictions with RR is to map the original feature vector to a higher dimensional space by applying a kernel function. There are kernels available which map the original input features into an infinite dimensional intrinsic feature space (e.g., gaussian kernel), then the

formulation with $\phi(x) \in \mathbb{R}^J$ is not applicable, due to the infinite character of it. [8]

A Kernel function $K(x_i, x')$ can be decomposed to the representative vector ($\phi(x) \in \mathbb{R}^J$) with $x' \in \mathbb{R}^M$ denoting a feature vector for which a prediction will be made. The Kernel function can also be interpreted as a similarity measure between two feature vectors. As mentioned above this decomposition can result in an infinite dimensional representative vector. [8]

$$K(x_i, x') = \phi(x)^T \phi(x') [8] \quad (2.83)$$

Luckily there is an option to utilize the kernel function to calculate the model parameters in an empirical space without transforming the features in the intrinsic space (this fact is known as the kernel trick). This is shown in Equation (2.89), it can be seen that all variables that are necessary for prediction map in the empirical space with N dimensions. [8]

Introducing a kernel function, the objective function for KRR can be expressed in the empirical space [8]:

$$J_{KRR}(a) = \|K^T a - y\|^2 + \rho \|\Phi(x) a\|^2 [8] \quad (2.84)$$

$$u = \Phi(x) a [8] \quad (2.85)$$

$$K = \Phi^T(x) \Phi(x) [8] \quad (2.86)$$

With $u \in \mathbb{R}^J$ denoting the intrinsic space model parameter vector, $a \in \mathbb{R}^N$ denoting the empirical space model parameter vector and $K \in \mathbb{R}^{N \times N}$ denoting the kernel matrix. [8]

Equation (2.84) & (2.85) & (2.86) leads to [8]:

$$J_{KRR}(a) = a^T K^2 a - 2a^T K y + \|y\|^2 + \rho a^T K a [8] \quad (2.87)$$

Equations (2.87) partial derivative with respect to $\mathbf{a} \in \mathbb{R}^N$ is set to zero (zero gradient), to find the analytical solution for $\mathbf{a} \in \mathbb{R}^N$: [8]

$$\mathbf{a} = [\mathbf{K} + \rho \mathbf{I}]^{-1} \mathbf{y} [8] \quad (2.88)$$

The following equation shows how to calculate a prediction ($\hat{y} \in \mathbb{R}^1$) in the empirical space: [8]

$$\hat{y} = f(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^T \mathbf{u} = \vec{\mathbf{k}}(\mathbf{x})^T \mathbf{a} = \sum_{i=1}^N a_i \tilde{\mathbf{K}}(\mathbf{x}_i, \mathbf{x}) [8] \quad (2.89)$$

And we denote that:

$$\vec{\mathbf{k}}(\mathbf{x}) = \boldsymbol{\Phi}^T(\mathbf{x}) \boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\phi}(\mathbf{x}_1)^T \boldsymbol{\phi}(\mathbf{x}) \\ \boldsymbol{\phi}(\mathbf{x}_2)^T \boldsymbol{\phi}(\mathbf{x}) \\ \vdots \\ \boldsymbol{\phi}(\mathbf{x}_N)^T \boldsymbol{\phi}(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{K}}(\mathbf{x}_1, \mathbf{x}) \\ \tilde{\mathbf{K}}(\mathbf{x}_2, \mathbf{x}) \\ \vdots \\ \tilde{\mathbf{K}}(\mathbf{x}_N, \mathbf{x}) \end{bmatrix} [8] \quad (2.90)$$

Let's point out that the prediction function (2.89) as well as the model parameter calculation (2.88) have only N dimensional or $N \times N$ dimensional variables, despite being able to predict infinite dimensional data caused by utilizing a kernel which is mapping to infinite dimensions. [8]

Summarized there are three steps necessary to build a Kernel Ridge Regression model and make a prediction with it: [8]

- Compute a symmetric kernel matrix $N \times N$ dimensional (Equation 2.86)
- Calculate the model parameters in the empirical space (Equation 2.88)
- Make predictions by computing $f(\mathbf{x})$ (Equation 2.89)

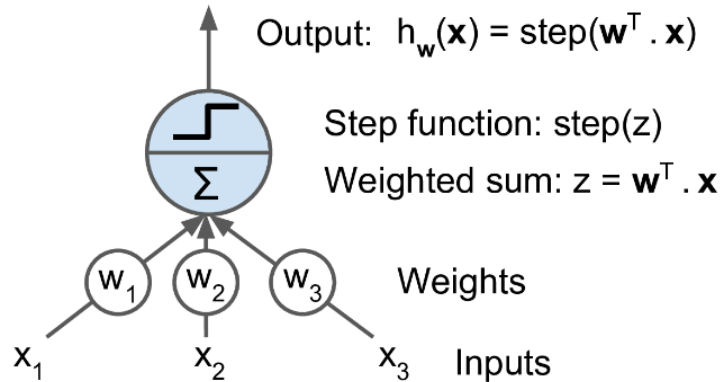


Figure 2.8: Visualization of a perceptron or neuron [4]

2.3.3 Dense Neural Network

Neural Networks are built up by neurons, also called perceptron's. A Perceptron calculates a weighted sum of its numerical inputs and was invented in 1957 by Frank Rosenblatt. The sum is fed to an activation function. The output of the perceptron is the value returned by the activation function. Figure 2.8 shows a single neuron with three inputs. [4]

A Multi-Layer-Perceptron combines multiple perceptron's to a connected network. It is built up by an input layer, an output layer, and multiple hidden layers. Such a network is called neural network and Figure 2.9 shows a simple neural network with one hidden layer. To the weighted sum computed in the neuron another term is added, which is called bias term. The bias is a constant numerical value for each neuron.

The neural network has the following parameters, they are called hyper parameters:

- Amount of hidden layers
- Neurons per hidden layer
- Activation functions
- Regularization

The hyper parameters can be tuned with proper optimization methods. Keras-Tuner is a python library that can perform such tunings of hyperpa-

rameters and is used in this thesis. There are several tuners available. In this work the Hyperband tuner is utilized. It is a derivative of a random search algorithm with implementations that speed up the tuning process. [9] [12]

A neural network has the following model parameters:

- Weights for each input into a neuron
- Bias for each neuron

The following objective function is minimized:

$$J_{DNN}(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - y_{pred})^2 + \rho ||\mathbf{w}||^2 \quad (2.91)$$

y_i is the groundtruth value, y_{pred} is the prediction, ρ is the regularization parameter and $\mathbf{w} \in \mathbb{R}^M$ is a vector that consists of the neural network weights where M is the amount of input streams that are weighted at the whole net.

The parameters are trained with an optimization algorithm called ADAM. It is a Stochastic Gradient Descent derivative which implements momentum-based methods and root mean square propagation. [6] [12]

Neural networks perform well on complex data for regression as well as classification problems and can tackle very large Machine Learning tasks with huge amounts of features. [4]

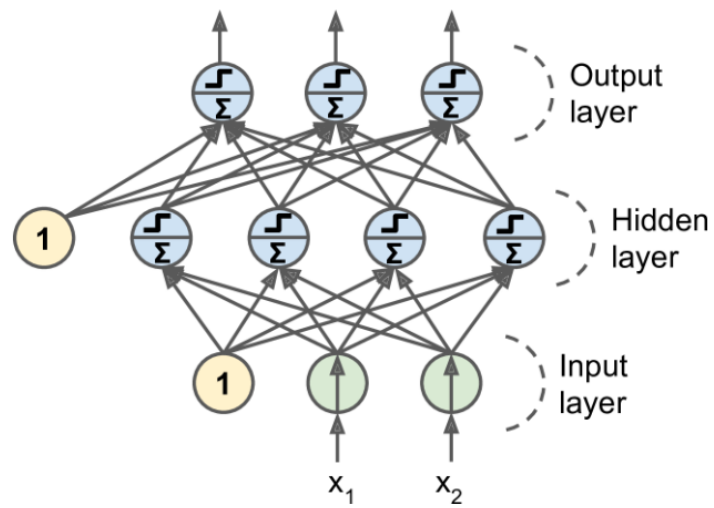


Figure 2.9: Neural network [4]

3 Practical Part

The focus of this chapter is to describe the two main concepts that have been processed in this thesis. At first the CFD simulation model is shown (3.1), the simulation model is utilized to generate a Fuel Cell Degradation dataset. Afterwards the Machine Learning pipeline is described (3.2), which utilizes the dataset from the simulations in order to create a prediction model.

3.1 CFD Simulation Model

In this chapter the CFD simulation model and setup are described. The first subchapter introduces the computational mesh grid and a qualitative view on the boundary conditions. In the boundary conditions subchapter, the quantitative values are shown that are applied at the boundaries. The numerical setup and the calibration methodology is described in the last two subchapters.

Fuel Cell degradation is simulated with the commercial Computational Fluid Dynamics software FireM©. The application performs calculations on a 3D discretized mesh, which is build up by finite volume elements. FireM© is a Multi-physics solver and has multiple modules implemented, the following are active during the simulations carried out in this work:

- Electromagnetic (2.2.2 & 2.2.3)
- Fuel Cell (2.2.3 & 2.2.4)
- Multiphase (2.2.3)
- Species Transport (2.2.3)
- Porosities

The electromagnetic module calculates all electric potentials and currents within the Fuel Cell. The Fuel Cell module handles the reaction kinetics of the hydrogen oxidation process as well as the degradation effects. To derive the gaseous, liquid, and solid-state properties and fractions, the multiphase module is activated. In order to keep track of the species concentrations, the species transport module is enabled. The porosity module is activated in the cooling channel domain to model a pressure drop in that domain. In the chapter 'FC-Balancing Equations' 2.2.3 the solved differential equations are described as well as the quantities that are solved with each equation.

The presented work is utilizing the solver version 2022.2.release.

3.1.1 Analysis Model

Simulations are performed on the AVL Geno Fuel Cell. Two models are investigated, a detailed model and a homogenized model.

The detailed model resolves all the geometrical features of the cell.

To minimize computational effort, the homogenized model is introduced. It resolves the complex and small structure of the gas feeding channels together with the solid bipolar plate in a simplified way. The mesh is overlaying both the solid and fluid locations. A fluid volume fraction determines how much space is associated to the fluid phase within the cells. Fluid permeabilities are utilized to model the pressure drops. The MEA details (layer thickness and physical properties) are resolved in a similar way as the resolved model. This is mandatory to simulate electrochemical effects as well as degradation.

The degradation simulations are carried out on the homogenized model.

Detailed model

The detailed model consists of a geometrical resolved mesh of all components of the Fuel Cell and is shown in figure 3.1. It is built up by a polyhedral finite volume mesh. In total it consists of 32.5 million cells. To protect the design of the AVL Fuel Cell, the figure showing the assembly is blurred.

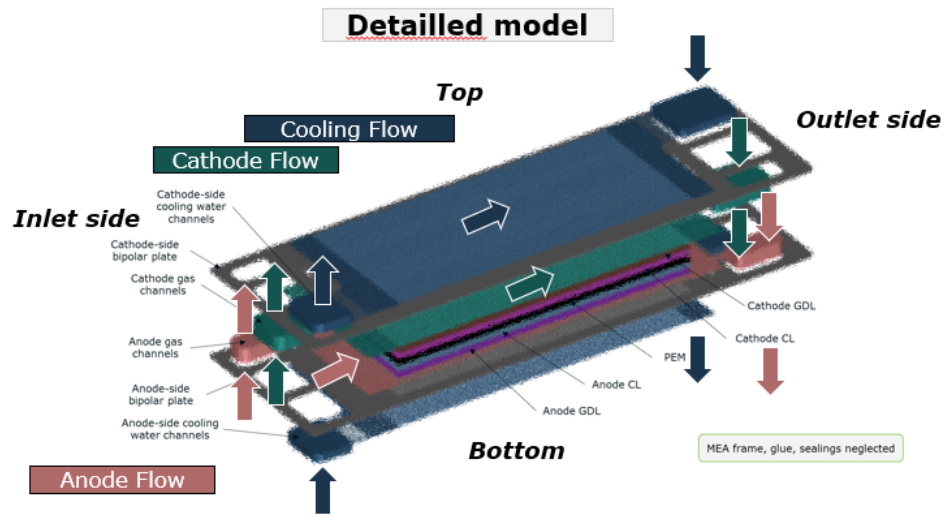


Figure 3.1: Detailed model

Note that the inclination of the fluid flows into the cell is modelled at the detailed mesh.

Fluid flow boundaries are represented by arrows. Arrows pointing inside the domain are inlets and arrows pointing outside the domain are outlets. The following boundary condition types are applied on the fluid model regions (**Cathode, Anode & Cooling channels**):

- Inlet side - Inlet: Mass flow
- Inlet side - Outlet: Mass flow
- Outlet side - Inlet: Mass flow
- Outlet side - Outlet: Absolute pressure

Homogenized model

The homogenized model is a simplified one, that has the requirement of keeping track of all the physics in a representative manner. This mesh is built up by hexahedron finite volume cells and consists of 76.870 cells. The simple mesh does not cover the inclination effects from the fluid channels

into the cell, in order to decrease the computational cell count. Therefore, each inlet and outlet boundary is aligned with the flow direction inside the cell.

The following boundary condition types are applied on the fluid model regions (**Cathode, Anode & Cooling channels**):

- Inlet: Mass flow
- Outlet: Absolute pressure

All walls to the environment are set adiabatic in both the detailed and the homogenized model. That implicates that all the heat, produced by the cell, is transported downstream via the cooling fluid. Note that two cooling flow paths are modelled, but just one MEA assembly.

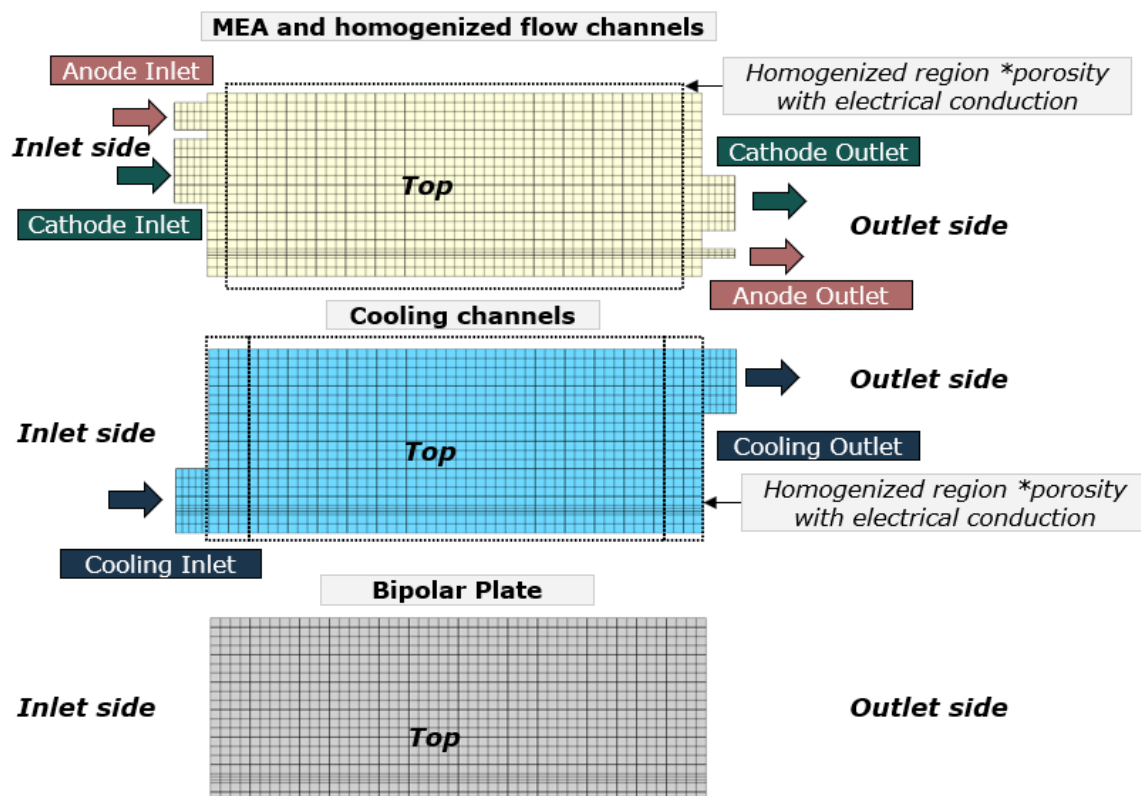


Figure 3.2: Homogenized model

3.1.2 Boundary Conditions

Three types of simulations are performed:

- Steady state simulations for fitting
- Steady state simulations to validate calibration
- Transient simulation to determine degradation effects

Calibration simulations

The boundary conditions of the calibration simulations are shown in table 3.1 & 3.2.

Name	Unit	Value	Value	Value	Value
Operating point		0.5A/cm ²	1.5A/cm ²	2.0A/cm ²	2.25A/cm ²
H ₂ - inlet,dry	mol%	78.5	87.7	88.5	87.7
Stoic	-	1.7	1.5	1.5	1.5
Pressure - outlet	bara	1.56	2.625	2.189	2.542
RH - inlet	%	78.4	90.5	92.2	93
Temperature - inlet	°C	60	60	60	60

Table 3.1: Anode boundary conditions - Calibration - Fuel

Name	Unit	Value	Value	Value	Value
Operating point		0.5A/cm ²	1.5A/cm ²	2.0A/cm ²	2.25A/cm ²
Stoic	-	2	1.8	1.8	1.8
Pressure - outlet	bara	1.225	2.248	1.853	2.108
RH - inlet	%	85	60	60	60
Temperature - inlet	°C	60	60	60	60

Table 3.2: Cathode boundary conditions - Calibration - Air

Validation simulations

To generate a polarization curve and compare it to measurements, the validation simulations are performed. The operating points 0.5A/cm² and 2.0A/cm² from the calibration simulations are utilized as well at the validation. The boundary conditions of the validation simulations are given in table 3.3 & 3.4.

All calibration and validation simulations have a mass flow of 41.97 kg/h for each cell, at 60°C, and atmospheric outlet pressure applied in the cooling domain.

Degradation simulations

The transient degradation simulations are performed on random generated parameter sets. The parameters and the upper and lower bounds of them are shown in table 3.5

3 Practical Part

Name Operating point	Unit	Value 0.1 A/cm ²	Value 0.3 A/cm ²	Value 1.0 A/cm ²	Value 1.5 A/cm ²
H ₂ - inlet,dry	mol%	71.7	77.4	86.9	88.3
Stoic	-	2.5	1.8	1.6	1.5
Pressure - outlet	bara	1.4	1.449	1.56	2.015
RH - inlet	%	86.8	77.7	79.1	80.7
Temperature - inlet	°C	60	60	60	60

Table 3.3: Anode boundary conditions - Validation - Fuel

Name Operating point	Unit	Value 0.1 A/cm ²	Value 0.3 A/cm ²	Value 1.0 A/cm ²	Value 1.5 A/cm ²
Stoic	-	5	3	1.8	1.8
Pressure - outlet	bara	1.085	1.121	1.224	1.688
RH - inlet	%	85	85	85	75
Temperature - inlet	°C	60	60	60	60

Table 3.4: Cathode boundary conditions - Validation - Air

In the simulation a load cycle is applied over a total simulation time of 500 hours. The load cycle has a trapezoidal shape that is shown in figure 3.3. Two different lengths of the cycle are simulated, 6 and 12 seconds. The shape of the load cycle is based on the accelerated stress test protocol from

Name	Unit	Lower Value	Upper Value
Ano H ₂ dry	mol%	70	90
Ano/Cat/Cooling in T	°C	50	70
Ano outlet P	bara	1.5	2.5
Ano RH	%	50	90
Ano Stoic	-	0.5	2.0
Cat outlet P	bara	1.5	2.5
Cat RH	%	50	95
Cat Stoic	-	0.5	2.5
Cooling mass flow	kg/h	21.6	43.2
Cooling P	bara	1.5	2.0
Maximum voltage	V	0.95	1.2
Minimum voltage	V	0.7	0.95

Table 3.5: Transient simulation parameter bounds

the US department of energy. [18] The material properties of the solid phase are set to the bipolar plate properties at all solid phases except the catalyst layers, the gas diffusion layers and the membrane, which are set to their specific properties.

3.1.3 Numerical Setup

All simulations, detailed as well as homogenized, are separated into three domains:

- MEA & anode/cathode flow channels
- Cooling water domain
- Bipolar plate domain

3 Practical Part

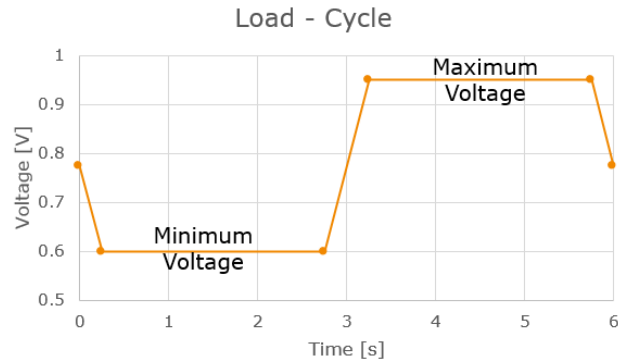


Figure 3.3: Simulation load cycle

Note that on the homogenized model the bipolar plate domain is smaller than in the detailed approach, due to the packing of solid parts into both fluid domains by utilizing porosities.

MEA & anode/cathode flow channels

This domain consists of three phases (gaseous, liquid, and solid) and has the following equations activated:

- Momentum and Continuity equation in the fluid phases (2.2.1)
- Energy equation (2.2.1)
- Multiphase volume fraction equations (2.2.3)
- Species equation at the gaseous phase (2.2.3)
- Dissolved water transport equation in the MEA (2.2.3)
- Dissolved species equation at the MEA in the solid phase (2.2.3)
- Platinum ion transport equation at the MEA in the solid phase (2.2.3)
- Electric potential equation in the solid phase (2.2.2)

The gaseous phase consists of CO_2 , H_2 , H_2O , H_2O_2 , HF , N_2 , and O_2 . The liquid phase is set to water and the solid phase is set to the material properties of the bipolar plate.

Equation	Differencing Scheme	Underrelaxation Value	Linear Solver
Continuity	Central	0.1	AMG
Momentum	AVL SMART Bounded	0.1	GSTB
Energy	Upwind	0.8	GSTB
Species	AVL SMART Bounded	0.9	AMG
Volume fraction	Upwind	0.8	AMG
Dissolved water	Upwind	0.5	AMG
Dissolved species	Upwind	1	AMG
Platinum Ion Transport	Upwind	1	AMG
Electric potential	Upwind	0.5	AMG

Table 3.6: Numerical setup: MEA & anode/cathode flow channels

Cooling water domain

The cooling water domain consists of two phases (liquid and solid) and has the following equations activated:

- Momentum and Continuity equation in the liquid phases (2.2.1)
- Energy equation (2.2.1)
- Multiphase volume fraction equations (2.2.3)
- Electric potential equation in the solid phase (2.2.2)

The liquid phase is set to water and the solid phase is set to the material properties of the bipolar plate.

Bipolar plate domain

The bipolar plate domain consists of one phase (solid) and has the following equations activated:

Equation	Differencing Scheme	Underrelaxation Value	Linear Solver
Continuity	Central	0.1	AMG
Momentum	MINMOD Relaxed	0.6	GSTB
Energy	MINMOD Relaxed	0.5	GSTB
Volume fraction	Upwind	0.8	GSTB
Electric potential	Upwind	0.5	AMG

Table 3.7: Numerical setup: Cooling water domain

- Energy equation (2.2.1)
- Electric potential equation in the solid phase (2.2.2)

Equation	Differencing Scheme	Underrelaxation Value	Linear Solver
Energy	MINMOD Relaxed	0.5	GSTB
Electric potential	Upwind	0.5	AMG

Table 3.8: Numerical setup: Bipolar plate domain

All domains utilize the SIMPLE algorithm to solve the Navier-Stokes-Equations, extrapolate the pressure to the boundaries and calculate the gradients by a least square fit method. The theoretical background to these concepts is well documented in literature [16].

3.1.4 Model Calibration

Both models (detailed and homogenized) are calibrated to a measured polarization curve. At the homogenized model an additional calibration is performed to fit the pressure drops to the measurement.

The target of the calibration process is to find a set of parameters in order to fit the performance in the simulation to measurements (polarization curve and pressure drops).

Polarization curve calibration

The fitting process to the polarization curve is necessary to capture several effects that are not modelled in the simulation. In particular the microstructure platinum distribution of the catalyst layers is not resolved, as well as the manufacturing and assembling characteristics of the complete MEA. Further information's to the electrochemical modelling procedure can be found in [3]. Table 3.9 shows the parameters that are calibrated to fit the polarization curve.

Name	Unit	Value Detailed	Value Homogenized
Charge transfer coefficient	-	0.6118	0.6002
Reference exchange current density	A/m ³	443	3137
Agglomerate radius	m	3 E-7	2 E-7
Ionomer film thickness	m	3 E-7	2 E-7

Table 3.9: Calibrated parameters of the polarization curve fitting

Figure 3.4 and 3.5 shows the comparison of measurement and simulation for detailed and homogenized model, which differ due to the resolution of the geometry.

Pressure drop calibration, homogenized model

The pressure drop calibration is done to guarantee that the homogenized model and the real application have an equal pressure field. In the homogenized model the geometry of the anode/cathode flow channels and the

3 Practical Part

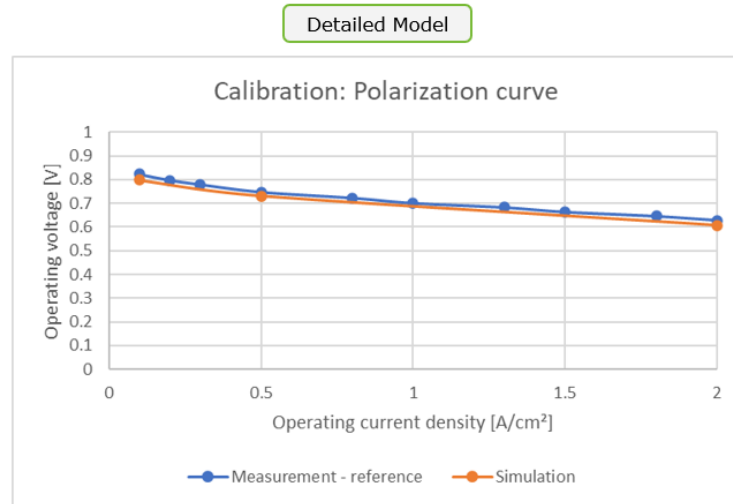


Figure 3.4: Calibrated polarization curve - detailed model

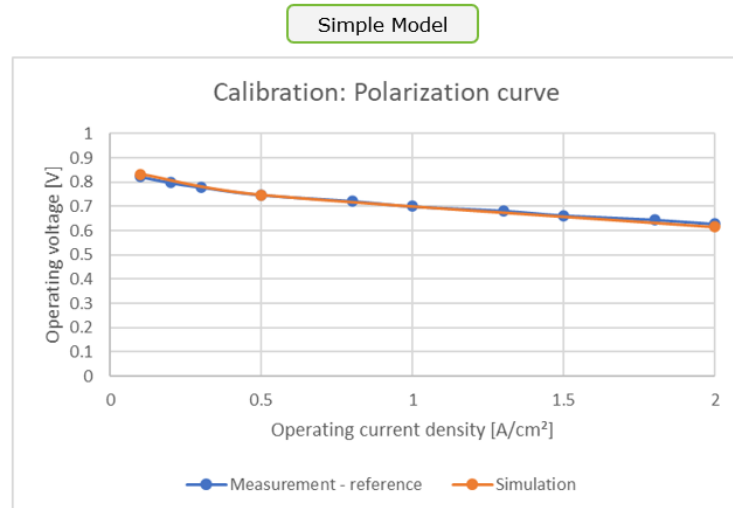


Figure 3.5: Calibrated polarization curve - homogenized model

bipolar plate is simplified by utilizing porosity models. The hindering of the fluid flow is modelled via fluid permeabilities. Further information's to the pressure drops modelling can be found in [3].

There is one fluid permeability value for each cartesian direction x, y and z.

Table 3.10 and 3.11 shows the calibrated values that are set in the performed simulations.

Name	Value Cell	Value Transition	Value Ports
Fluid permeability - x	8.5 E-10	4.3 E-10	4.3 E-9
Fluid permeability - y	4.3 E-14	4.3 E-9	4.3 E-9
Fluid permeability - z	8.5 E-10	4.3 E-10	4.3 E-9

Table 3.10: Calibrated fluid permeabilities anode

Name	Value Cell	Value Transition	Value Ports
Fluid permeability - x	2.5 E-9	1.2 E-9	1.2 E-8
Fluid permeability - y	1.2 E-13	1.2 E-8	1.2 E-8
Fluid permeability - z	2.5 E-9	1.2 E-9	1.2 E-8

Table 3.11: Calibrated fluid permeabilities cathode

Note that on the cathode the focus was set to calibrate the pressure drops at low current densities (high voltage in the cell) and drawbacks are made in the bigger current densities/flow rates region. The two curves above are not connected due to different stoichiometric ratios which would give a wrong impression because the pressure drop is mainly related to the flow rate.

3 Practical Part

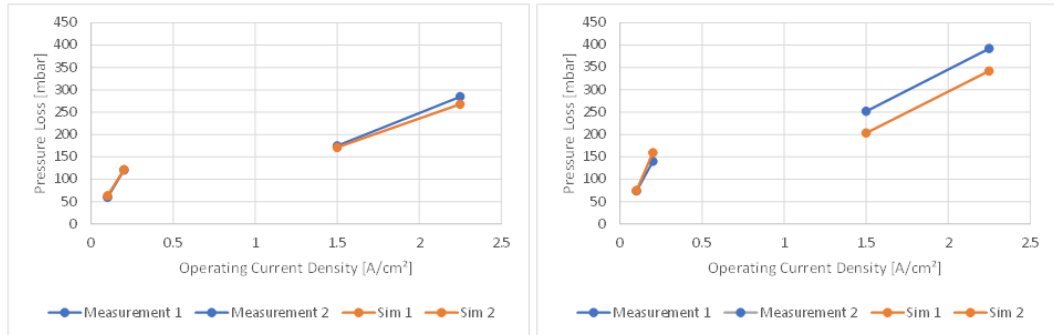


Figure 3.6: Calibrated pressure drop - anode (left) and cathode (right)

However, this representation is given to show an realistic pressure drop at several current densities or in other words at different operating points.

3.2 Machine Learning Model

In this chapter the machine learning model, that is used to predict degradation quantities, is described. At first the data pre-processing is presented (3.2.1). It covers the way from the raw input data into features that are suitable to feed into a machine learning pipeline. Afterwards the dataset strategy (3.2.2) is shown. The data is split in several portions for model comparison, hyperparameter tuning and generalization testing. At chapter 3.2.3 the build-up models are described with their specific hyperparameter setup.

3.2.1 Data Preprocessing

The raw data consists of features/inputs and labels/outputs. The target of the pre-processing phase is to transform the features into distributions that are similar to a standard distribution, because ML-models can process such data accurately. This chapter is separated into the feature and label pre-processing.

Feature Pre-processing

The dataset consists of 16 scalar features and one time series. The scalar features describe several operating conditions that are assumed constant over the fuel cell runtime. The time series describes the load profile which is applied to the fuel cell.

All features, the scalars as well as the load profile features, are scaled with a MinMax-Scaler before feeding them into the ML pipeline. Due to the random generation of the scalar features within certain ranges no other feature transformation steps are necessary. Details to the scaler can be found in the Scikit-Learn documentation. [14]

The load profile is de-featured with three different approaches:

- Class approach
- Principal Component Analysis
- Adapted Principal Component Analysis

Lets start with the simple **class approach**. At this method the operating voltage range is separated into several classes. Each class has an associated interval of voltage. e.g., class-i covers 0.6-0.7 V. One count represents one data-point of the time series that lays in the class specific range. The total count of each class is fed to the machine learning pipeline.

Figure 3.7 shows a load profile with 6 samples. The resulting features are the following:

- Class 1 : 2 counts
- Class 2 : 0 counts
- Class 3 : 2 counts
- Class 4 : 2 counts

Both **PCA approaches** are following the same ideas but differ in the treatment of the trainings curves, the differences are discussed later in this chapter. At these approaches the time series data is transformed into weights, they correspond to the principal components of the PCA model. The number of principal components defines the number of extracted features from the time series. Further information to PCA is given in chapter 2.3.1.

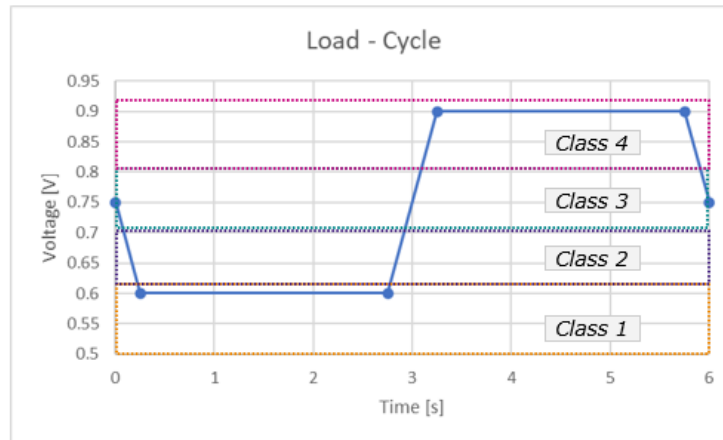


Figure 3.7: Example of the class approach decomposition

At first generic time series data is generated to train a PCA model. The following curve types are generated for training purposes:

- Trapezoidal time series - figure 3.8
- Triangular time series - figure 3.9
- Piecewise constant time series - figure 3.10

Within this work the following curve parameters are selected:

- Maximum length: 1.800.000 seconds
- Minimum length: 540.000
- Maximum voltage 1.1 V
- Minimum voltage: 0.5 V

Note that the model is designed to process test-bed data (mostly accelerated stress tests) as well as real automotive fleet data (piecewise constant character and low frequent cyclic data - depending on the driving route)

Trapezoidal curves with random cycle lengths between 5-10 seconds are generated with a probability of 25% and are implemented due to the similarity to accelerated stress testing protocols. [18]. In real automotive applications such high frequencies in load change are typically not the case, but cyclic load changes at lower frequencies. That's why trapezoidal curves with cycle

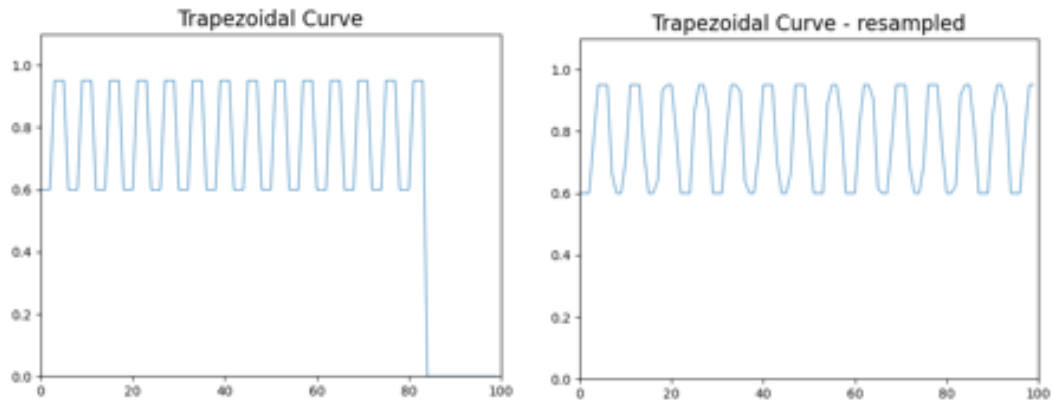


Figure 3.8: Example of trapezoidal curve data

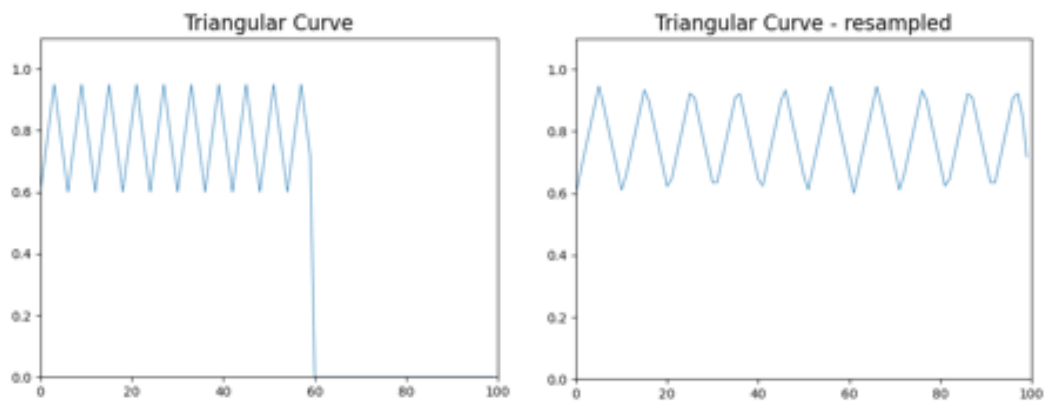


Figure 3.9: Example of triangular curve data

3 Practical Part

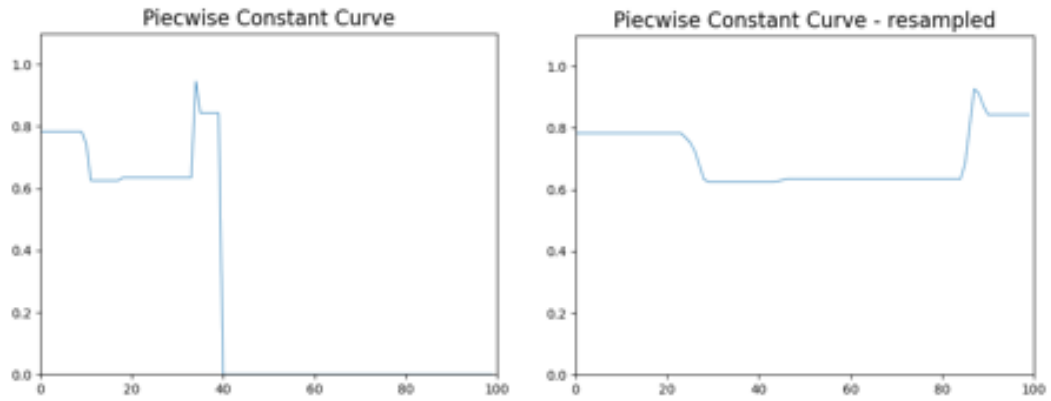


Figure 3.10: Example of piecewise constant curve data

lengths between 50-150 seconds are implemented with a probability of 30%. Another curve type associated to accelerated stress tests is a triangular shape. They are implemented with a probability of 15% and cycle lengths between 4-10 seconds. Piecewise constant curves are created with a probability of 30% and a random amount of load changes ranging from 0-100.

Note that for PCA each time series data, that is processed with a PCA model, needs to consist of the same total sample count. As mentioned, there are two different PCA approaches investigated in this work. The first one, from now on called PCA approach, fills up the generated curves of all types with zeros to end up with curves that have all the same sample count (length). The second approach, from now on called adapted PCA approach, performs an interpolation on all curve types to guarantee that they have the same sample count (length).

Figures 3.8 & 3.9 & 3.10 show all curve types for both approaches, but with smaller length to see their characteristics. On the left are curves associated to the PCA approach and on the right to the adapted PCA approach.

And finally, the time series, that should be de-featured, is transformed with the trained PCA models. The result is a weight vector that describes the weight of each principal component for the transformed time series. The weight vector is fed to the machine learning models.

Label Preprocessing

Seven labels are predicted with the prediction models. It is derived by manually testing MinMax-Scaler, Standard-Scaler, linear manipulations, and logarithmic transformation. Standard Scaler is documented in the Scikit-Learn documentation. [14] The following pre-processing strategy is used in this work:

- Label 1: MinMax
- Label 2: MinMax (0.1-0.9); Log; MinMax (0-1)
- Label 3: Log; MinMax
- Label 4: Log; MinMax
- Label 5: Log; MinMax
- Label 6: Log; MinMax
- Label 7: Log; MinMax

Figure 3.12 shows the distributions of the transformed labels and 3.11 shows the distribution of the labels before transformation.

3.2.2 Dataset/Training Strategy

The dataset is separated into several portions for different purposes. The following portions are extracted from the dataset.

- Training dataset
- Validation dataset
- Test dataset

The training dataset and the validation dataset is utilized to determine a hyperparameter set for both the Kernel Ridge Regression model and the Dense Neural Network model.

The Kernel Ridge Regression workflow performs hyperparameter tuning with the Scikit-Learn library (GridSearchCV). The function determines the best parameters with the concept of cross validation. That means that the data is separated into portions of same size, in this work the k-fold method is utilized with a value of five. Therefore, five portions are extracted from

3 Practical Part

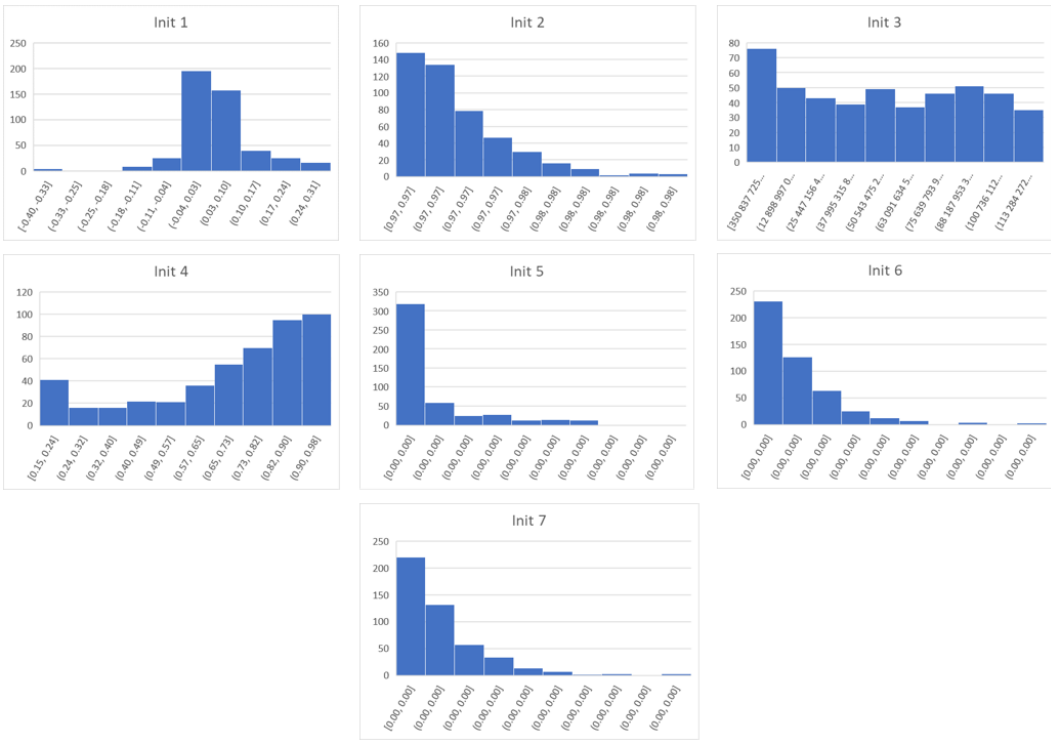


Figure 3.11: Initial Labels

3.2 Machine Learning Model

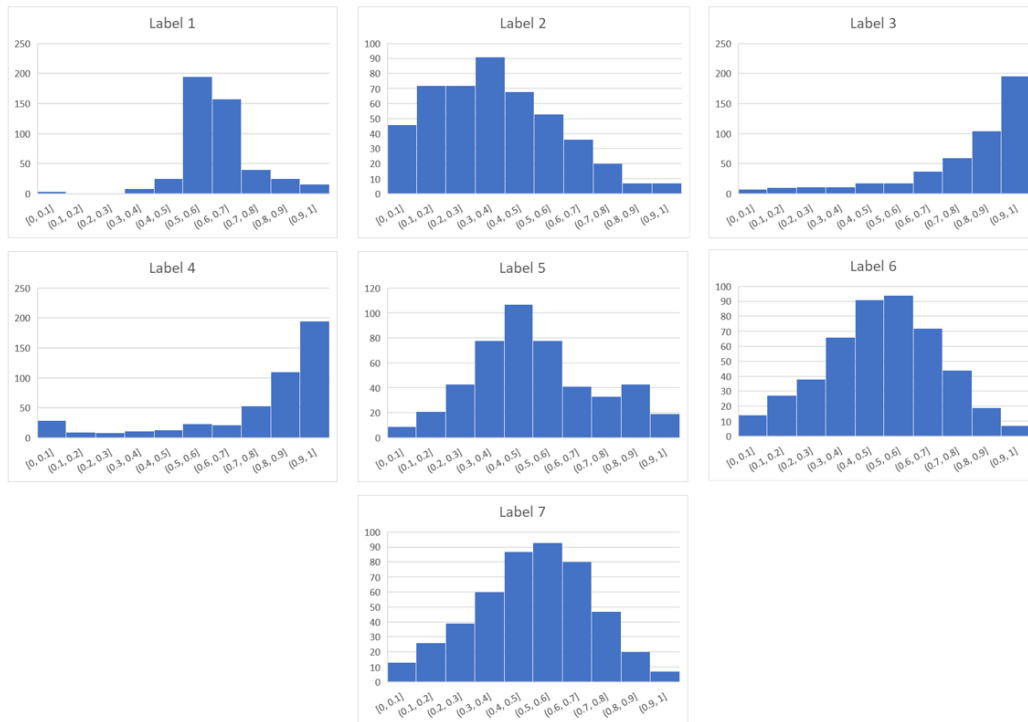


Figure 3.12: Transformed Labels

the data, four for training and one for validation. That is repeated five times so that each portion is once taken for testing of the prediction model.

The Dense Neural Network workflow performs hyperparameter tuning with the Keras-Tuner and TensorFlow libraries. At first a parametrized model builder function is defined. Afterwards the Hyperband function is called to determine a suitable set of hyperparameters. The validation dataset is used to calculate the performance of the derived models from the model builder function.

The second step is model evaluation. At this step the model's performance are compared against each other with their derived hyperparameters set. Due to the usage of the validation dataset at the complete tuning process, at the Dense Neural Network workflow, this data is not used again for model evaluation. Both models are trained and evaluated on the training dataset with k-fold Cross Validation and a value of five.

At the last step both models are trained with the training dataset and the validation dataset. Afterwards the model generalization ability is determined with the test dataset. Generalization means how good the models perform on data they have not seen during training. Note that both models never seen the test dataset before and that the complete data is shuffled upfront separation.

To summarize three steps are performed at this work:

- Hyperparameter tuning
- Model evaluation
- Generalization evaluation

Figure 3.13 illustrates the workflow described in this chapter.

3.2.3 ML Model Setup

In this chapter the build-up models are described. At first an overview of the hyperparameter space and the models architecture is presented. Afterwards the best model parameters are shown.

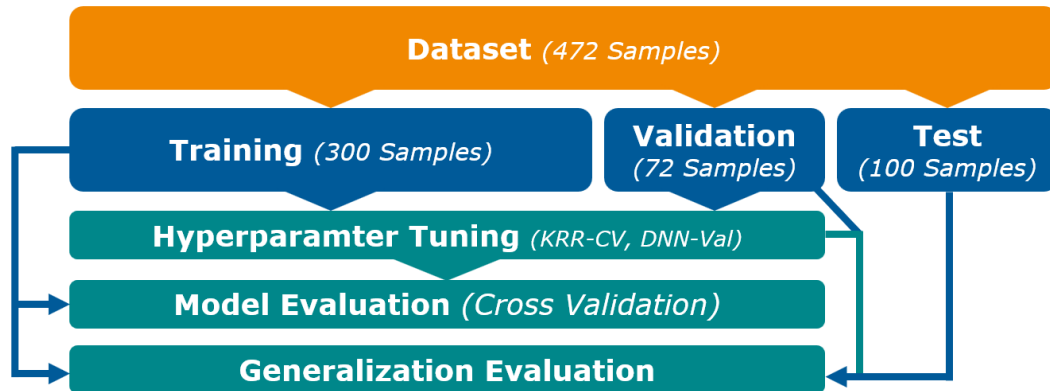


Figure 3.13: Dataset strategy

Kernel Ridge Regression

The Regression model utilizes a Kernel and Ridge Regularization. The Kernel function enables the model to predict non-linear behaviour even though the Regression model is a linear one. To keep the model parameters of the Regression small, a Ridge Regularization term is added at the training procedure. Further information is presented in chapter 2.3.2.

The following Hyperparameter space is set during tuning:

- Kernel type (RBF, Laplacian, Sigmoid)
- Kernel parameter (γ) ($E-6$ - $E+6$ in $E+1$ steps)
- Kernel parameter (only at sigmoid) ($coef0$) ($E-6$ - $E+6$ in $E+1$ steps)
- Regularization parameter (α) ($E-12$ - $E+6$ in $E+1$ steps)

The Kernels and their parameters are well described in the Scikit-Learn documentation. [14] Note that each KRR model is trained to predict just one label. Therefore seven models are trained, one for each label.

The following set of hyperparameters was derived during hyperparameter tuning.

3 Practical Part

Name	Label 1	Label 2	Label 3	Label 4	Label 5	Label 6	Label 7
Kernel	RBF	Laplace	Laplace	Laplace	Laplace	Laplace	Laplace
γ	0.1	0.01	0.1	0.1	0.1	0.01	0.01
<i>ceof0</i>	-	-	-	-	-	-	-
α	1 E-6	1 E-11	1 E-12	1 E-9	1 E-11	1 E-5	1 E-5

Table 3.12: KRR parameters - Class Approach

Name	Label 1	Label 2	Label 3	Label 4	Label 5	Label 6	Label 7
Kernel	RBF	RBF	Laplace	Laplace	Laplace	Laplace	Laplace
γ	0.1	0.1	0.01	0.1	0.001	1E-5	1E-5
<i>ceof0</i>	-	-	-	-	-	-	-
α	1 E-6	1 E-5	1 E-12	1 E-11	1 E-11	1 E-12	1 E-12

Table 3.13: KRR parameters - PCA Approach

Name	Label 1	Label 2	Label 3	Label 4	Label 5	Label 6	Label 7
Kernel	RBF	Laplace	Laplace	Laplace	Laplace	Laplace	Laplace
γ	0.1	0.01	0.1	0.1	0.1	0.001	0.001
<i>ceof0</i>	-	-	-	-	-	-	-
α	1 E-6	1 E-11	1 E-10	1 E-12	1 E-11	1 E-10	1 E-11

Table 3.14: KRR parameters - PCA adapted Approach

Dense Neural Networks

The Dense Neural Network consists of a linear input layer, three to six hidden layers and an linear output layer. All layers utilize Ridge Regularization to keep the weights small. Linear layer means that the activation function passes the value through without transformation through an activation function. Several activation functions are considered during tuning in the hidden layers. The training procedure aims to minimize the mean squared errors of the training data prediction to the ground truth.

Summarized the following Hyperparameter space is set during tuning:

- Hidden Layers (3 - 6)
- Neurons per Layer (Amount of labels - 6*Amount of Features)
- Activation Function (Relu, Sigmoid, Elu, SoftMax)
- Regularization parameter ($E-11$ - $E-3$ in $E+1$ steps)

A detailed description of the activation functions is well documented in the TensorFlow documentation [1].

The following parameters are determined during hyperparameter tuning:

Name	Class Approach	PCA Approach	PCA-adapted Approach
Hidden Layers	5	4	6
Neurons Layer 1	62	217	112
Neurons Layer 2	102	102	172
Neurons Layer 3	67	92	147
Neurons Layer 4	37	197	177
Neurons Layer 5	107	-	132
Neurons Layer 6	-	-	147
Activation Function	Relu	Relu	Relu
Regularization parameter	1 E-5	1 E-6	1 E-5

Table 3.15: DNN parameters

4 Results

This chapter provides an overview of the results generated within this work. There are two main categories of results, CFD simulation outcome and post-processing of the Machine Learning model.

The CFD results are separated into three chapters. The first one will cover the comparison between the detailed and the homogenized model (4.1) to get an impression of the quality loss due to simplification. The second CFD results chapter is presenting a comparison between an Accelerated-Stress-Test on the test-bed and the virtual CFD simulation (4.2). And finally results of the generated dataset, based on the homogenized model, are presented (4.3).

The Machine Learning model post-processing is divided into a model comparison (4.4) and a prediction post-processing (4.5). The model comparison is showing results for the Kernel-Ridge-Regression model and the Dense-Neural-Network. Afterwards the KRR model is utilized to derive trend analysis, e.g., what happens if the anode stoichiometry is varied? These trends are post-processed in the prediction trend analysis chapter.

4.1 Comparison detailed vs. homogenized simulation

The comparability from the detailed model to the homogenized model is presented in this chapter. The post-processing is separated into the following parts of the fuel cell:

- Cathode flow channels
- Anode flow channels

4 Results

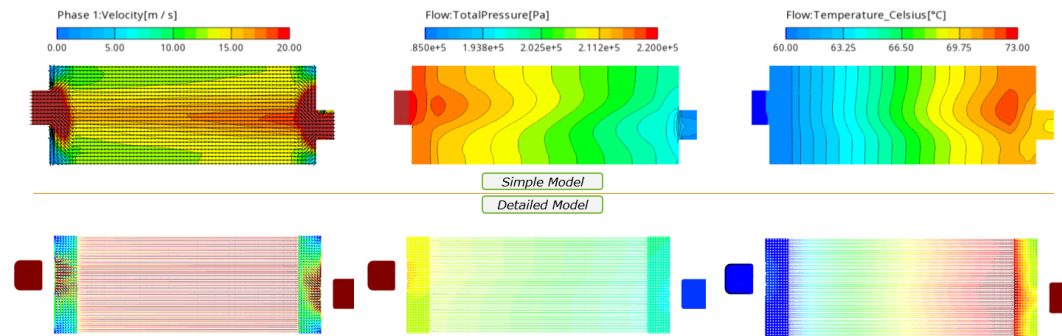


Figure 4.1: Comparison cathode flow: Velocity - Total Pressure - Temperature

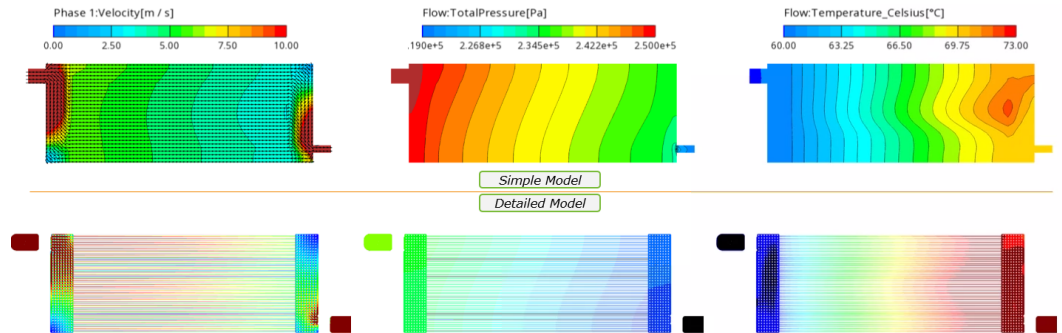


Figure 4.2: Comparison anode flow: Velocity - Total Pressure - Temperature

- Coolant flow channels
- Cathode catalyst layer
- Membrane

Both cathode and anode flow field (Figure 4.1 & 4.2) shows a good comparability with drawbacks in the temperature distribution (homogenized 3°C cooler). The detailed model predicts less pressure loss then the measurements show. That's why the anode pressure levels do not match between the homogenized and the detailed model, due to the calibration of the simple model to replicate measurements.

4.1 Comparison detailed vs. homogenized simulation

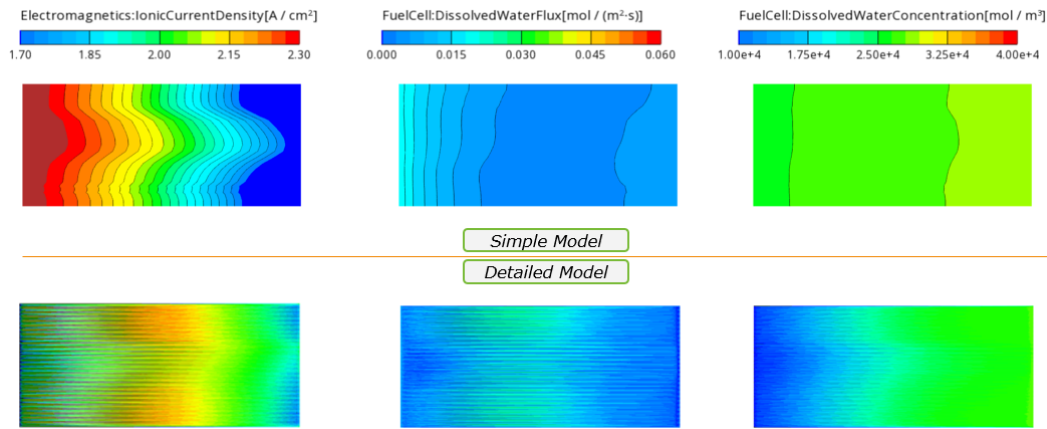


Figure 4.3: Comparison membrane: Ionic current density - dissolved water flux - dissolved water concentration

Results of the membrane are shown in Figure 4.3. The distribution of the ionic current density, water flux and water concentration show differences between the detailed and the homogenized approach, but the order of magnitude is comparable.

In the cooling domain the pressure and velocity levels correspond very well between the simple and detailed model (figure 4.4). The temperature distribution, predicted by the detailed model, does not match with the homogenized approach. Especially the gradient between cathode and anode side cannot be reproduced. But this gradient is questionable and could be produced by the model setup. Several cells are stacked upon each other and in the simulation model just one MEA domain is modelled but two cooling domains. The homogenized model is dissipating the heat more homogeneous across the simulation domain.

4 Results

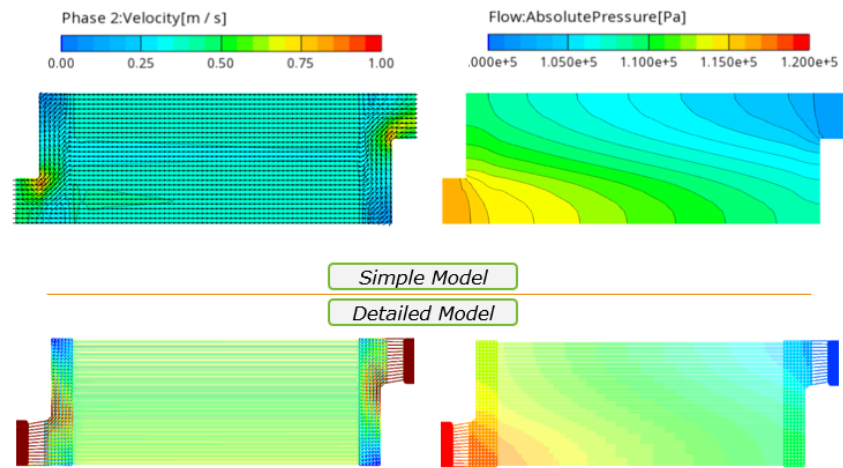


Figure 4.4: Comparison coolant flow: Velocity - Absolute Pressure

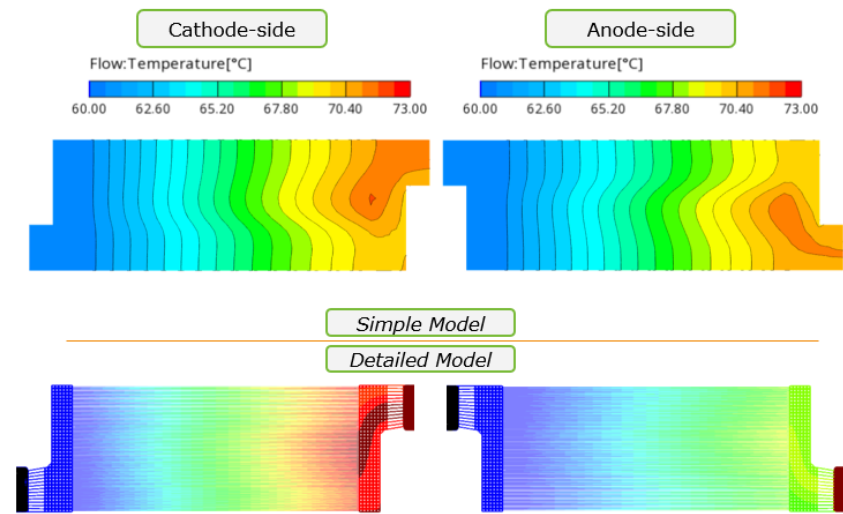


Figure 4.5: Comparison coolant flow: Temperature on cathode and anode side

4.1 Comparison detailed vs. homogenized simulation

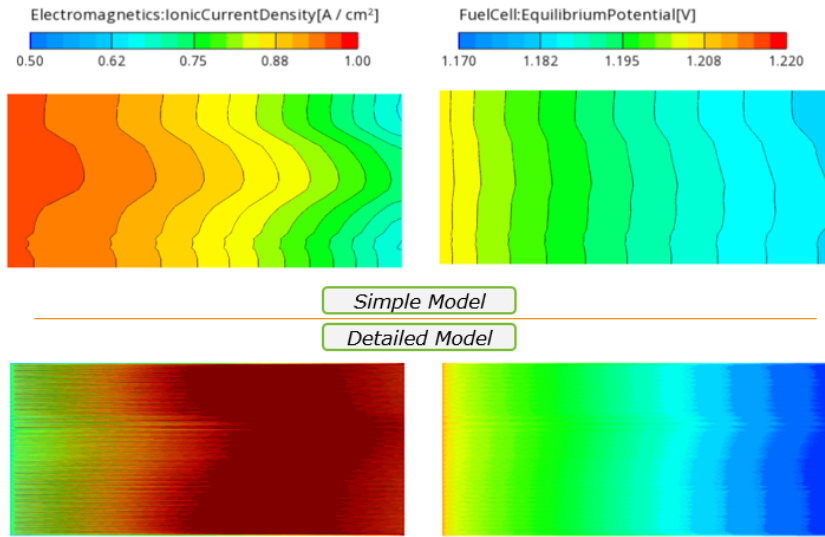


Figure 4.6: Comparison cathode catalyst layer: Ionic current density - equilibrium potential

In the cathode catalyst layer, the equilibrium potential matches well and the distribution of the ionic current density show discrepancies. The discrepancy in the ionic current density can be explained with a simplified setup in the homogenized model in which the electrical conductivity was set isotropic, but due to the cell design the conductivity is anisotropic. The detailed model captures this effect better than the homogenized model does. The quality of the simple model can be enhanced by taking anisotropic material properties into account .

Degradation effects are mostly affected by the temperature, humidity (water concentrations) and local potential, and the overall order of magnitude of these quantities correspond to each other but there are drawbacks in the three-dimensional distribution.

Summarized, the simulation results show a compromise on local results due to the simplification and anisotropic electrical conductivity in the homogenized model.

Main trends due to operating conditions should be captured due to the similarity of degradation relevant quantities.

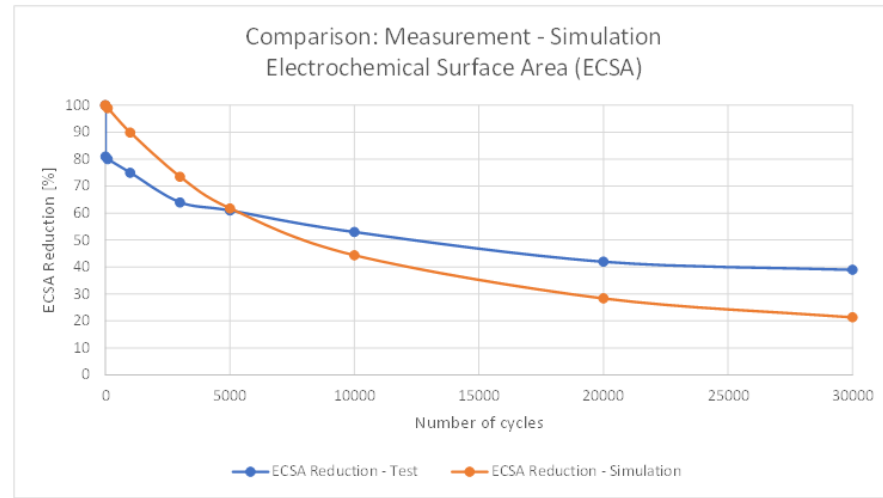


Figure 4.7: ECSA loss Comparison: simulation vs. measurement

4.2 Comparison simulation vs. measurement

This chapter shows a comparison between simulation (homogenized model) and measurement on the investigated cell. Results are based on the electrocatalyst accelerated stress test, provided by the DOE test protocols. A trapezoidal load profile is applied and repeated during this protocol. The maximum voltage is 0.95V and the minimum load is 0.6V. One cycle has a duration of 6 seconds. The experiment runs 50 hours and afterwards the reduction of the electrocatalytic surface area is measured. [18]

Figure 4.7 shows that the simulation model predicts more degradation than the measurement shows. Measurement shows a ECSA reductions of 60% and simulation suggests 80%. The measurement has a major drop in the first few cycles. This trend was not seen during simulation. This discrepancy could be optimized by fine tuning the model parameters in further calibration loops.

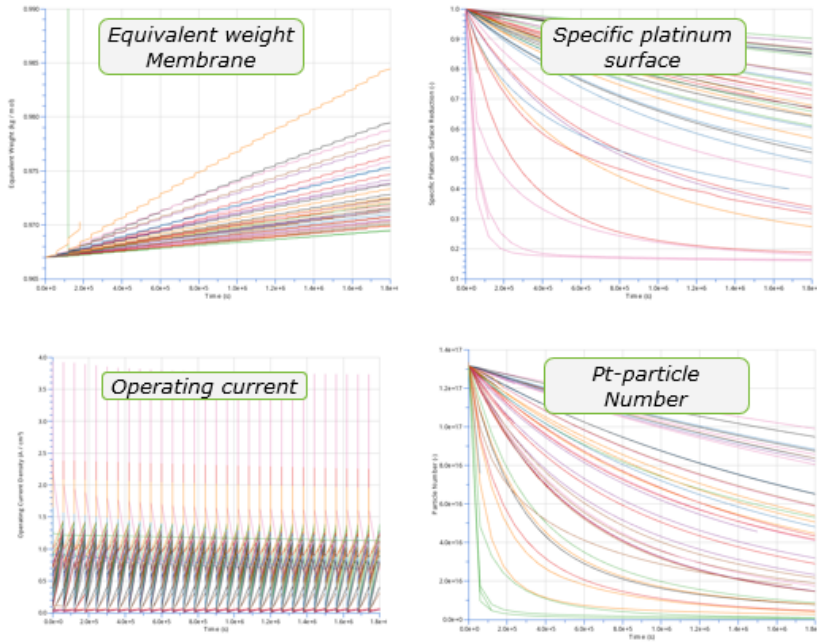


Figure 4.8: Degradation trends on the homogenized model

4.3 Homogenized model

This chapter shows the simulation results of the homogenized model that are utilized as labels for the Machine Learning models. The equivalent weight is an indicator for the ionic conductivity in the membrane. Specific platinum surface and platinum particle number describe the ECSA evolution of the cathode catalyst layer. The thicknesses show the shrinking of the MEA due to degradation and the operating current describes the performance loss of the cell.

The 2D degradation results show a widespread over the widespread operating conditions, these effects and trends will be processed by the machine learning model.

4 Results

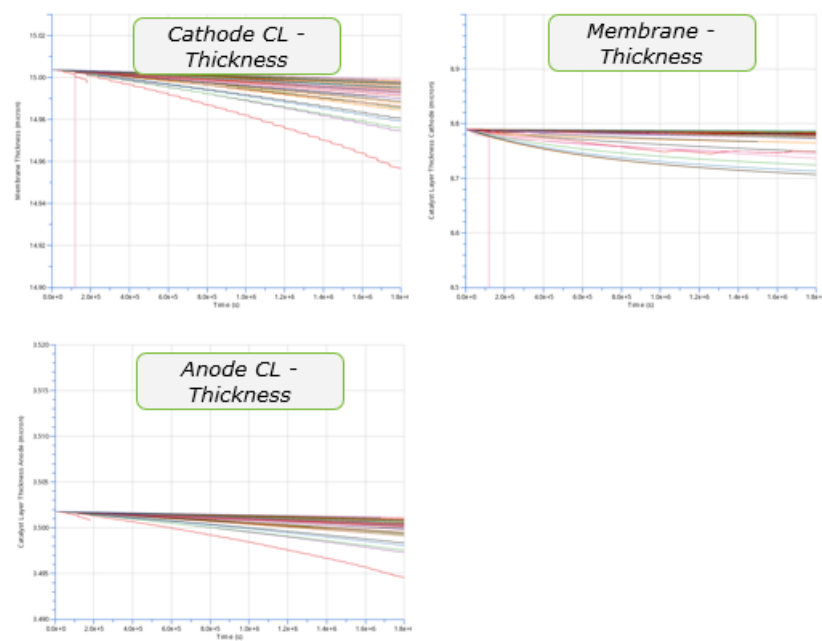


Figure 4.9: 2D Results of the homogenized model

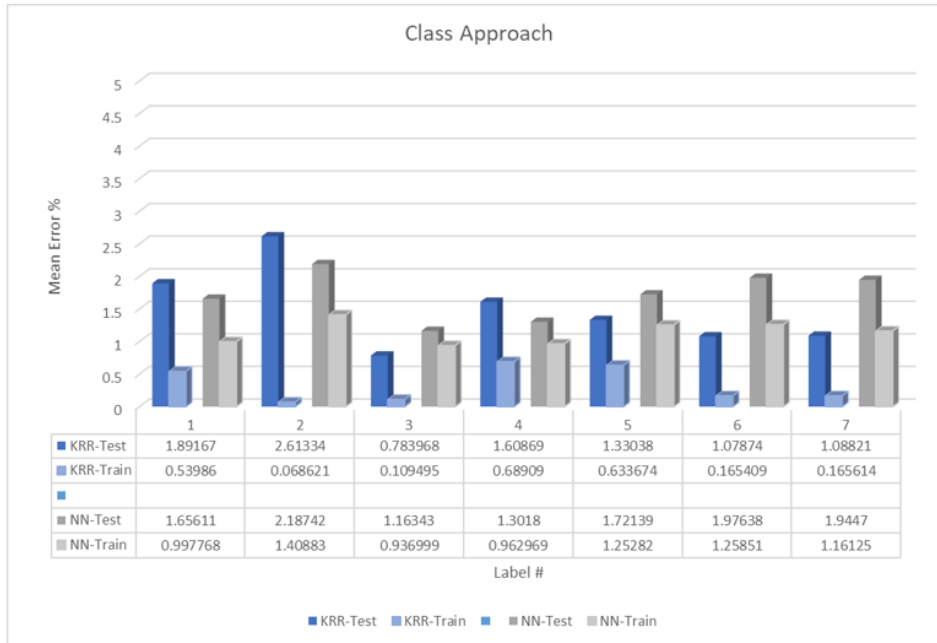


Figure 4.10: MAPE - Class approach

4.4 Machine learning model comparison

In this chapter the machine learning models are compared with a focus on the prediction errors.

Figure 4.10 & 4.11 & 4.12 show the Mean Average Percentage Error (MAPE) of the Machine Learning prediction models. The blue bars show Kernel Ridge Regression (KRR) results, and the grey bars show Dense Neural Network (DNN) results. The dark accent describes the error on the test dataset and the bright accent describes the errors on the training dataset. The horizontal axis describes the label on which the error is evaluated. The coloured boxes, at the PCA and adapted PCA results, show if the performance has increased (green), decreased (red) or stayed the same (yellow) compared with the class approach.

Overall, a low average error $<4\%$ is produced with each model at the feature prediction. The KRR model performs better compared to the DNN approach.

4 Results

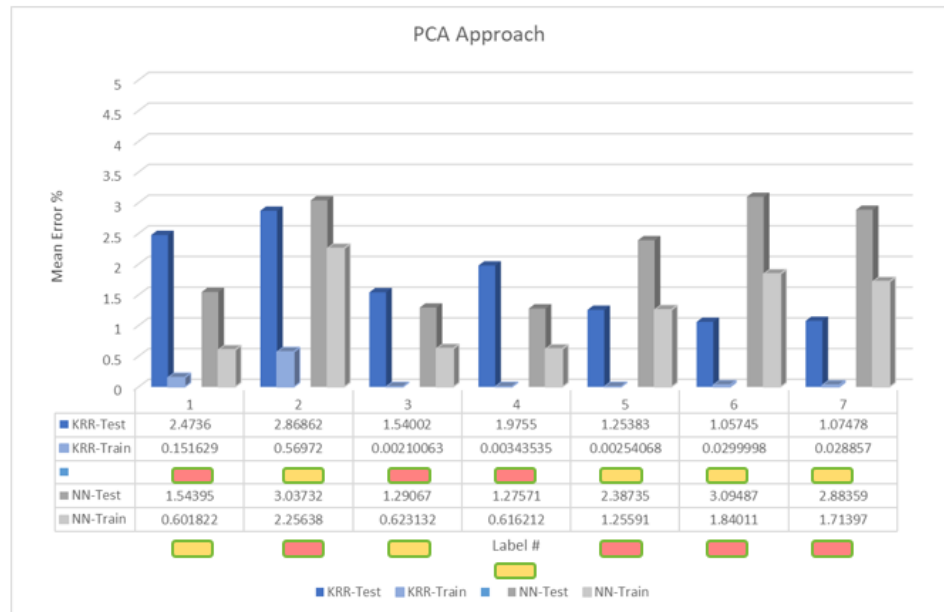


Figure 4.11: MAPE - PCA approach

Note that the discrepancy between the trainings and test errors are smaller at the DNN model, this indicates a better capability of predicting new feature sets (generalization). The class approach has the best prediction accuracy, but the adapted PCA approach is just slightly worse.

4.4 Machine learning model comparison

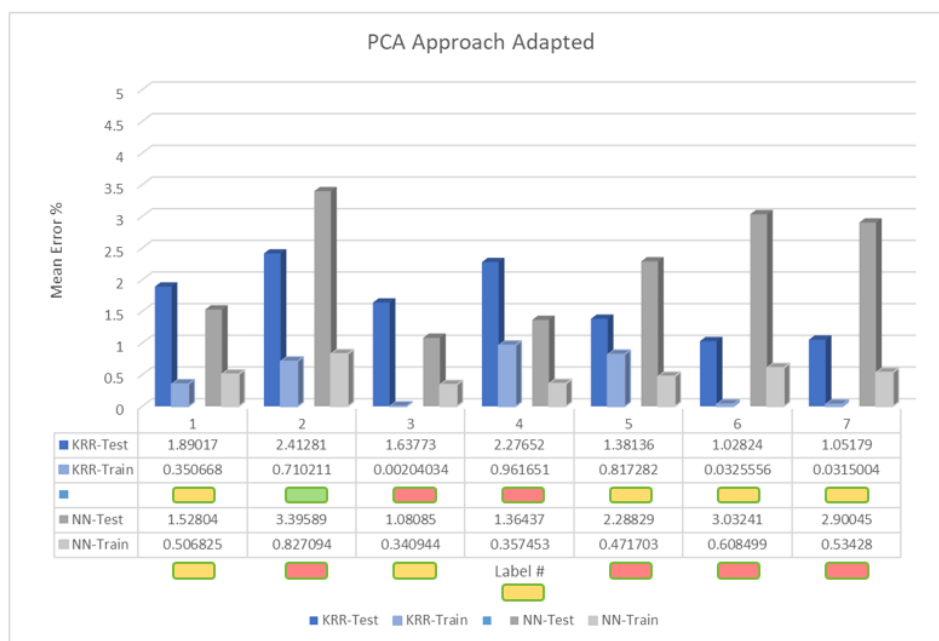


Figure 4.12: MAPE - PCA adapted approach

4 Results

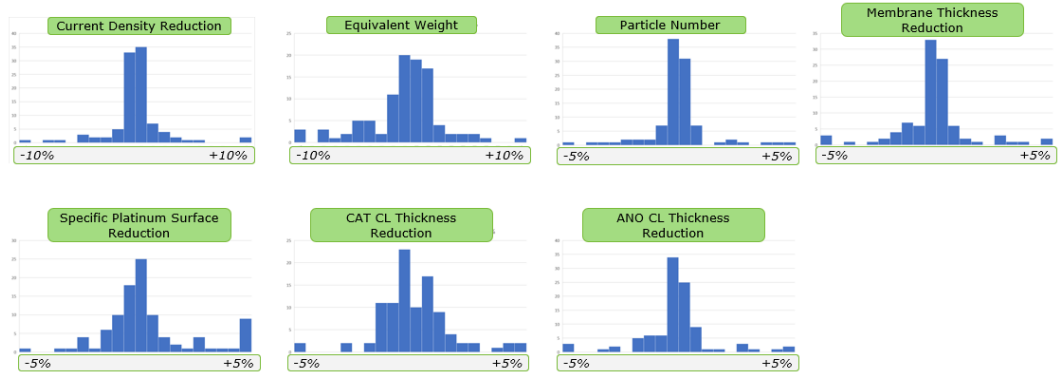


Figure 4.13: MAPE distribution - KRR - Class approach

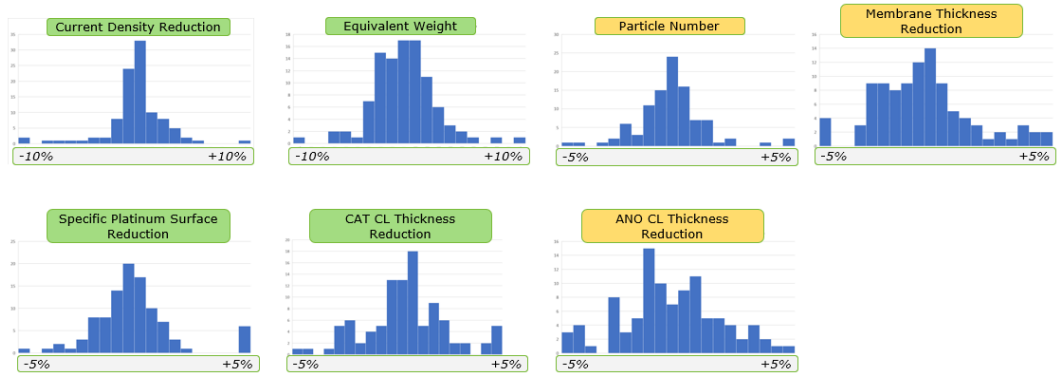


Figure 4.14: MAPE distribution - DNN - Class approach

Figure 4.13 & 4.14 & 4.15 & 4.16 show the distribution of the MAPE errors on the test dataset for each datapoint.

The best distribution of errors is achieved with the KRR model and the class approach, to de-feature the load profile (figure 4.13).

Note that the KRR model with the adapted PCA approach also performs very good (figure 4.15). The DNN model performs worse for both the class and the PCA approach. (figure 4.16 & 4.14)

4.4 Machine learning model comparison

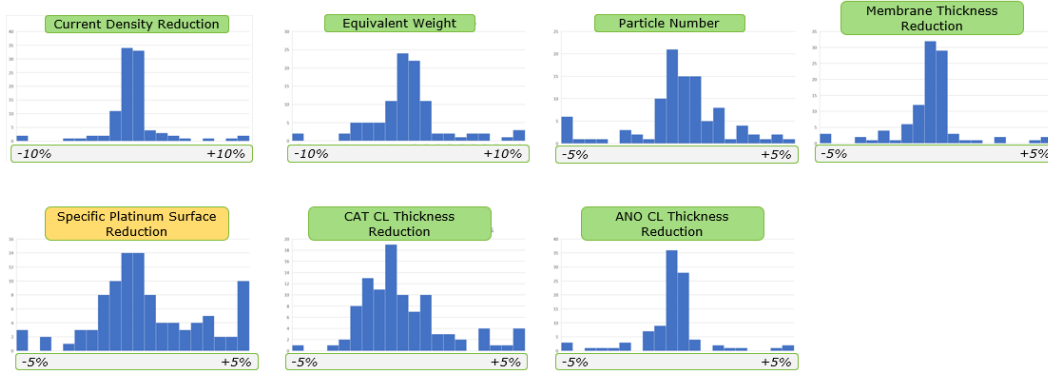


Figure 4.15: MAPE distribution - KRR - PCA adapted approach

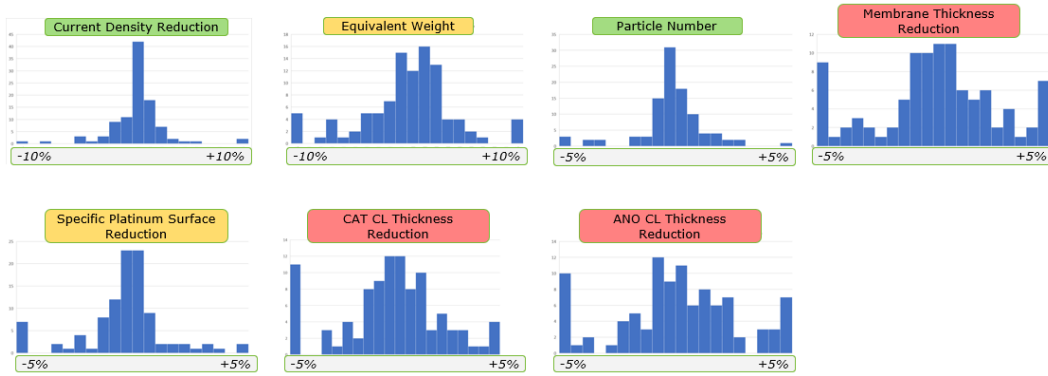


Figure 4.16: MAPE distribution - DNN - PCA adapted approach

The KRR model with the PCA adapted approach is identified as the favoured prediction model. On the one hand side this model shows low prediction errors and on the other hand side the potential of capturing the load profile characteristics is bigger on the PCA approach. This hypothesis is driven by the idea that the principal components can keep track of the load profile specific shape and not only of the load magnitude, like the class approach.

4 Results

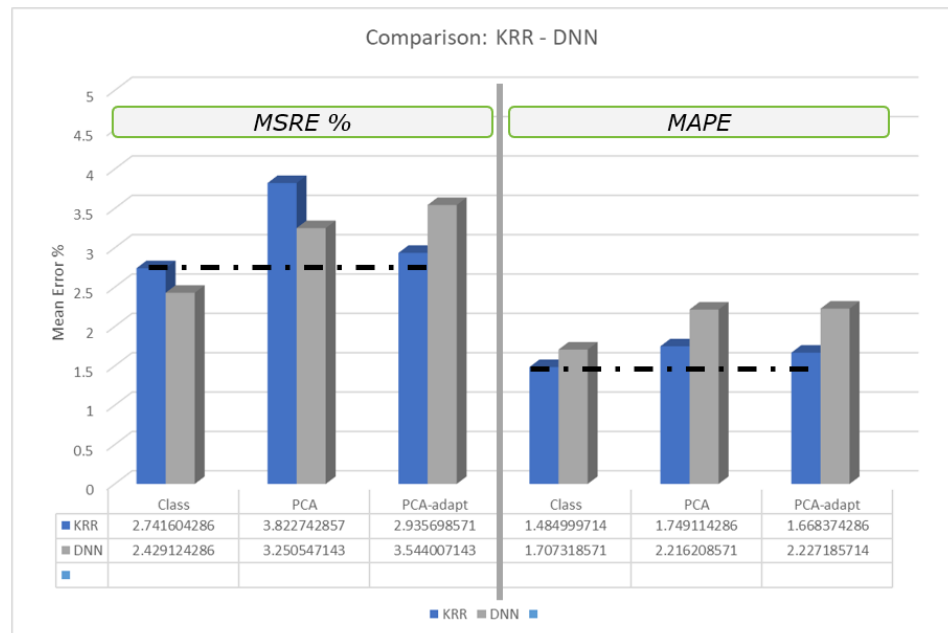


Figure 4.17: Summary of prediction accuracy

A summary of the averaged error over all labels is presented in Figure 4.17. The quadratic error (left side) and the linear error (right side) is shown for the test dataset and each ML model as well as each load profile de-feature method. The comparison of the linear and squared error at the class approach shows that the linear error is smaller at the KRR model compared with the DNN model, but the squared error is bigger at the KRR model. That indicates that the KRR model has bigger outlier at the prediction deviations then the DNN model. This effect is not seen at the adapted PCA approach.

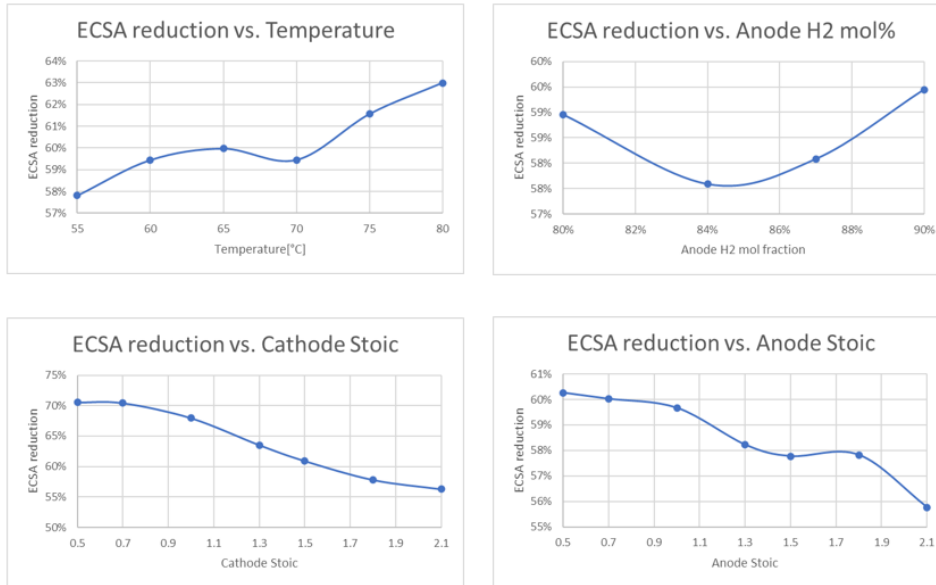


Figure 4.18: Trend analysis

4.5 KRR - prediction trend analysis

Finally, the KRR prediction model with the PCA adapted approach was trained with the complete simulation dataset. Afterwards trends were generated by varying specific features. Figure 4.18 shows the effect of temperature, anode H₂ mol fractions, cathode stoichiometry and anode stoichiometry on the ECSA loss.

The results show that the model captures non-linear dependencies by utilizing a kernel function, as expected.

A big influence of the cathode stoichiometry is predicted by the model. The variation of the cathode stoichiometry between 0.5-2.1 results in a ECSA loss between 55%-70%. A variation of temperature, H₂ mol fraction and anode stoichiometry results in ECSA differences <5%.

5 Outlook

The introduced Machine Learning models are able to predict Fuel Cell degradation effects, based on CFD simulation data, with average errors <4%. The simulation model overestimates specific platinum surface area reduction by 30%.

To increase the accuracy of the Machine Learning models and the CFD simulation models, the following aspects could be investigated:

CFD simulation model:

- Refined calibration of the homogenized model to the detailed model (electrical conductivities, thermal conductivities, porosity settings)
- Calibration of degradation model parameters in order to fit measurements
- Model half of the cooling domain on both sides of the MEA and apply a periodic boundary condition to capture the cooling effects more realistic
- Perform more simulations on several load profile types (triangular cycles, constant voltage, longer cycle times)

Machine Learning model:

- Add features to make several cell architectures comparable (geometric features, material features)
- Implement more operating conditions as time series (e.g. temperature, pressure)
- Process more data sources with the workflow (e.g. other simulation tools, AST-tests, fleet data)
- Perform a detailed feature importance analysis to check the PCA de-feature workflow

5 Outlook

The derived Machine Learning models of this thesis can be utilized to generate fast parameter variations to learn about the component's sensitivity to specific operating conditions and load cycles. Furthermore, that knowledge can be implemented into operating strategies to increase components lifetime.

Bibliography

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] A.o.Univ.-Prof. DI Dr. Raimund Almbauer, Ass.Prof.DI Dr. Michael Lang, and DI Dr. Alexander Trattner. Advanced thermodynamics, 2021.
- [3] AVL-List-GmbH. AST-Documentation-FireM, 2022.
- [4] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition, 2017.
- [5] Univ.-Prof. Dr.-Ing. habil. Günter Brenn and Ass.-Prof. Dipl.-Ing. Dr.techn. Walter Meile. Strömungslehre und Wärmeübertragung 1, 2015.
- [6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [7] Ambroz Kregar, Gregor Tavcar, Andraz Kravos, and Tomaz Katrasnik. Predictive system-level modeling framework for transient operation and cathode platinum degradation of high temperature proton exchange membrane fuel cells. *Applied Energy*, 263(C), 2020.

- [8] S. Y. Kung. *Kernel Methods and Machine Learning*. Cambridge University Press, 2014.
- [9] Giulia DeSalvo Afshin Rostamizadeh Ameet Talwalkar Lisha Li, Kevin Jamieson. Hyperband: A novel bandit-based approach to hyperparameter optimization.
- [10] Wolfgang Nolting. *Grundkurs Theoretische Physik 3: Elektrodynamik*. Springer-Lehrbuch. Springer Berlin Heidelberg, Berlin, Heidelberg, 10. aufl. 2013 edition.
- [11] Ryan P O’Hayre. *Fuel cell fundamentals*. Third edition. edition, 2016.
- [12] Tom O’Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. Kerasuner. <https://github.com/keras-team/keras-tuner>, 2019.
- [13] Arun Pandey, Zhiwei Yang, Mallika Gummalla, Vadim V. Atrazhev, Nikolay Yu Kuzminykh, Vadim I. Sultanov, and Sergei Burlatsky. A carbon corrosion model to evaluate the effect of steady state and transient operation of a polymer electrolyte membrane fuel cell. *Journal of The Electrochemical Society*, 160(9):F972–F979, 2013.
- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [15] Steven G. Rinaldo, Patrick Urchaga, Jingwei Hu, Wendy Lee, Jürgen Stumper, Cynthia Rice, and Michael Eikerling. Theoretical analysis of electrochemical surface-area loss in supported nanoparticle catalysts. *Phys. Chem. Chem. Phys.*, 16:26876–26886, 2014.
- [16] A.o.Univ.-Prof. DI Dr. Wolfgang Sanz. Computational fluid dynamics for compressible flows, 2021.
- [17] Ass. Prof. Dipl.-Ing. Dr.techn. BSc. Vanja Subotic. Brennstoffzellentechnik, 2022.
- [18] U.S. DRIVE Fuel Cell Tech Team. Accelerated stress test and polarization curve protocols for pem fuel cells, April 2023.

- [19] K.H. Wong and Erik Kjeang. Macroscopic in-situ modeling of chemical membrane degradation in polymer electrolyte fuel cells. *Journal of the Electrochemical Society*, 161:F823–F832, 05 2014.