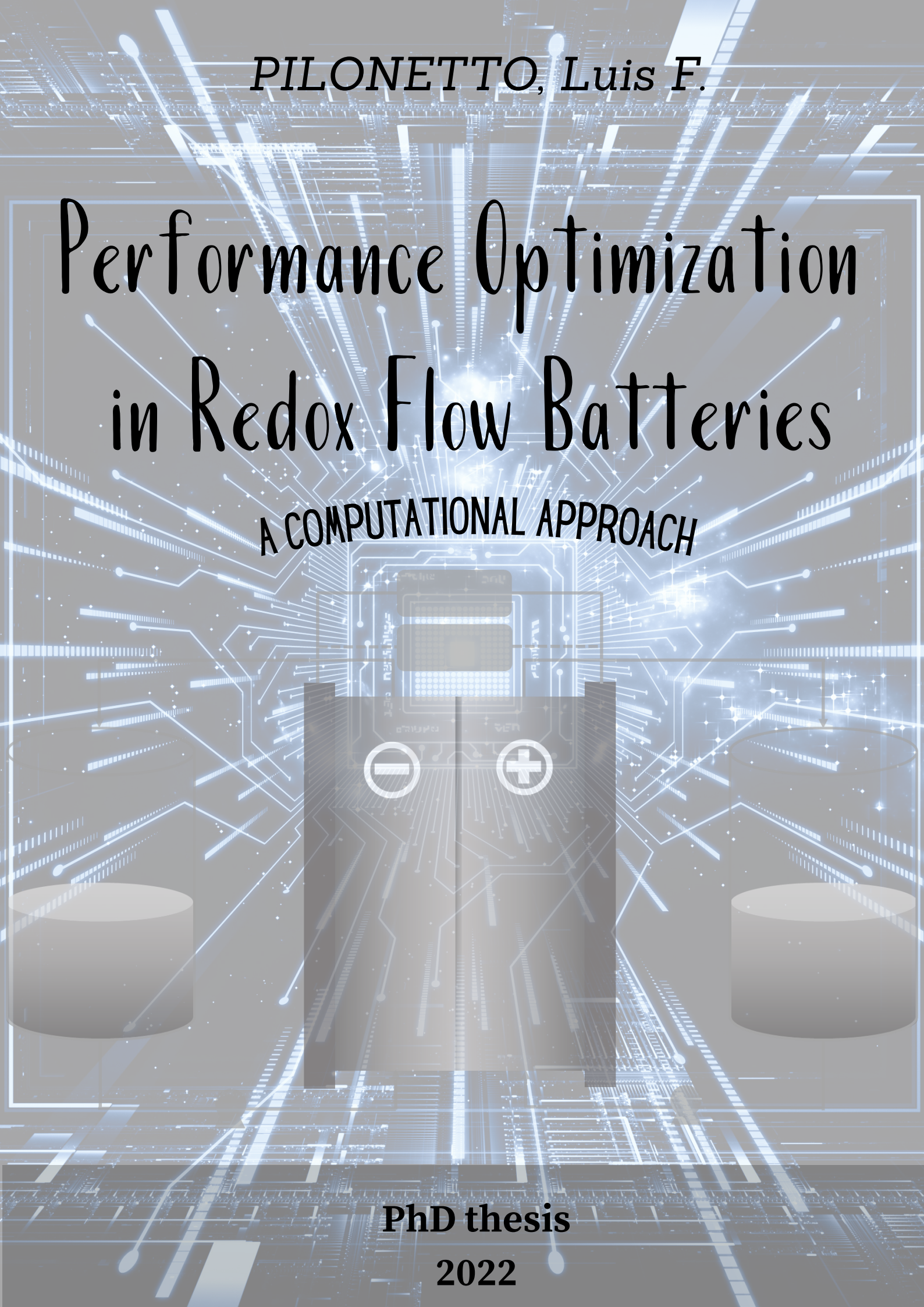
PILONETTO, Luis F.

Performance Optimization in Redox Flow Batteries

A COMPUTATIONAL APPROACH



PhD thesis

2022

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE QUÍMICA
PROGRAMA DE PÓS-GRADUAÇÃO EM QUÍMICA

Performance Optimization in Redox Flow Batteries: A Computational Approach

Luis Felipe Pilonetto*

Tese apresentada como parte dos requisitos para obtenção do título de DOUTOR EM CIÊNCIAS, área de concentração: FÍSICO-QUÍMICA

Orientador: Prof. Dr. Ernesto Chaves Pereira
*Bolsista FAPESP

São Carlos – SP
2022

Pilonetto, Luis Felipe

Performance Optimization in Redox Flow Batteries: A Computational Approach / Luis Felipe Pilonetto -- 2022.
234f.

Tese de Doutorado - Universidade Federal de São Carlos, campus São Carlos, São Carlos
Orientador (a): Prof. Dr. Ernesto Chaves Pereira de Souza
Banca Examinadora: Profa. Dra. Lúcia Helena Mascaro Sales, Prof. Dr. Emerson Rodrigues de Camargo, Profa. Dra. Mariana de Souza Sikora, Prof. Dr. Hudson Giovani Zanin
Bibliografia

1. Redox Flow Batteries. 2. Electrochemistry. 3. Finite Element Method. I. Pilonetto, Luis Felipe. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática (SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Ronildo Santos Prado - CRB/8 7325

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Química

Folha de Aprovação

Defesa de Tese de Doutorado do candidato Luis Felipe Pilonetto, realizada em 22/02/2022.

Comissão Julgadora:

Prof. Dr. Ernesto Chaves Pereira de Souza (UFSCar)

Profa. Dra. Lúcia Helena Mascaro Sales (UFSCar)

Prof. Dr. Emerson Rodrigues de Camargo (UFSCar)

Profa. Dra. Mariana de Souza Sikora (UTFPR)

Prof. Dr. Hudson Giovani Zanin (UNICAMP)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Química.

"[…] It takes over my energy. […] If the trunk was rotting, this thing that fed on it was now too strong to let it go. […]"

Andrew Solomon

"Because science is inseparable from the rest of human endeavor, it cannot be discussed without making contact, sometimes glancing, sometimes head-on, with a number of social, political, religious and philosophical issues."

Karl Sagan

Dedication

This thesis is dedicated to my mother, who showed me how life can be hard and has taught me to be better every day. I love you!

Unfortunately, I have to finish my thesis one year and a half before the predicted time because of health issues. But I think I took advantage of this live experience. I have learned a lot with professors, with friends, with science, with my disease, and with myself. This path has shown me how science works and how tough is to become a scientist in Brazil nowadays. I have enormously proud of all the people who make science in this country.

I am especially grateful to my advisor, Prof. Dr. Ernesto Chaves Pereira, and my former advisor, Profª. Drª Mariana Sikora, who played a key role in my recovery and the conclusion of my Ph.D. Thanks for the emotional and scientific support!

I am grateful to my co-advisor, Prof. Dr. Marco Antônio Barbosa Ferreira, who gave me a different and critical point of view about science and made me like organic chemistry (a tough one!).

I am grateful to my friends, Choko, Ellen, and Luana, who even far away were always close in the happy and the tough moments. Thank you, guys!

Further, I am grateful to all my relatives who always believed in me and supported my dreams.

Lastly, I am grateful for the friends I have made in this time.

Acknowledgments

I am especially grateful to the São Paulo Research Foundation (FAPESP, Fundação de Amparo à Pesquisa do Estado de São Paulo) grant 2019/11095-9, which financed this study.

I am grateful to the Federal University of São Carlos (UFSCar, Universidade Federal de São Carlos), its Graduate Program in Chemistry (PPGQ, Programa de Pós-Graduação em Química), the Coordination for the Improvement of Higher Education Personnel (CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and the Center for the Development of Functional Materials (CDMF, Centro de Desenvolvimento em Materias Funcionais).

List of Abbreviations

BB	*Box-Behnken*
BE	*Battery Efficiency*
CAES	*Compressed Air Energy Storage*
CC	*Central Composite*
CE	*Coulombic Efficiency*
CV	*Cyclic Voltammetry*
D	*Doehlert*
EDG	*Electron Donating Group*
EE	*Energy Efficiency*
EES	*Energy Storage System*
EWG	*Electron Withdrawing Group*
FEM	*Finite Element Method*
HER	*Hydrogen Evolution Reaction*
IFF	*Interdigitated Flow Field*
LSV	*Linear Sweep Voltammetry*
MAE	*Mean Absolute Error*
NHE	*Normal Hydron Electrode*
OCP	*Open Circuit Potential*
ODE	*Ordinary Differential Equation*
OER	*Oxygen Evolution Reaction*
PDE	*Partial Differential Equation*
RFB	*Redox Flow Battery*
RMSE	*Rooted Mean Squared Error*
ROM	*Redox Organic Molecule*
SFF	*Serpentine Flow Field*
SHE	*Standard Hydrogen Electrode*
SoC	*State of Charge*
SS	*Squared Sum*
VE	*Voltage Efficiency*
VRFB	*Vanadium Redox Flow Battery*
WR	*Weighted Residual*

Nomenclature

a specific surface area

ad additional geometric parameter

c concentration

d diameter

i density current

i unit value of current (+1 for charging and -1 for discharging)

j unit value of half-cell representation (-1 for negative and +1 for positive)

k rate constant

n number of electrons, quantity (in mols), number of…

p pressure

q charge

t time

u ionic mobility

v velocity

w weight

z valence

C capacity

D diffusion coefficient

E energy density or Nernst potential

F Faraday's constant

H length of residence path

I current

L thickness

N flux

O oxidized species

P power

Q charge

R universal gas constant or reduced species

S source of species

T temperature

U theoretical cell potential

V cell potential, volume

Greek

α transfer coefficient, pump efficiency, angle of trapezoid geometry

β angle of trapezoid geometry, degree of dissociation

δ thickness of membrane/electrolyte interfacial region

ε porosity of the electrode

η overpotential

θ angle of modification

κ permeability

μ viscosity

ρ density

σ conductivity

ω volumetric flow

ϕ potential

Subscript

$0i;\ i \in N$ first-order coefficient or effect

0 first

1 initial condition

2 second

a activation

act active species

$appl$ applied

br between reservoir

c charge procedure or concentration

cc current collector

d discharge procedure, dissociation

e electrode

f fiber of porous electrode

fix fixed

i species i

$ii;\ i \in N$ second-order coefficient

$ij;\ i \neq j; i,j \in N$ interaction coefficient or effect

in inlet electrode channel

l liquid or ionic

loc local

m membrane

max maximum

min minimum

neg negative electrode

o ohmic

out outlet electrode channel

ox oxidized species

p peak

p pump

pos positive electrode

red reduced species

s solid or electronic

t total

tp from trapezoid geometry

CP central point value

avg average value

br between reservoirs (or tanks)

g geometric

p parametric

var variable

opt optimum

Superscript

0 standard

b bulk

cc current collector

e electrolyte or electrode

eff effective value

er electrolyte interfacial region

j negative or positive electrode

$junc$ junction

m membrane

mr membrane interfacial region

n negative electrode

p positive electrode

s surface

t electrolyte tanks

y, x species charge

List of Tables

List of Figures

Resumo

OTIMIZAÇÃO DE DESEMPENHO EM BATERIAS DE FLUXO REDOX: UMA ABORDAGEM COMPUTACIONAL.

As baterias de fluxo redox (RFBs) surgem como uma tecnologia alternativa para o armazenamento de energia em larga escala. As principais características deste tipo de dispositivo estão relacionadas ao seu design único, que proporciona escalabilidade, desacoplamento entre potência e densidade de energia e a utilização de diversos tipos de espécies ativas. A RFB mais desenvolvida até hoje é a bateria de fluxo redox de vanádio (VRFB), que usa $V^{2+/}V^{3+}$ em uma semi-célula e VO^{2+}(V(IV))/VO_2^{+}/(V(V)) em outra. Apesar de suas vantagens, a VRFB enfrenta alguns desafios relacionados ao desempenho, o que dificulta sua penetração no mercado. Dois dos problemas mais relevantes são a perda de capacidade e a perda de voltagem. A perda de capacidade é causada pela seletividade não ideal da membrana, que permite a contaminação cruzada entre as semi-células, levando a reações de auto-descarga. A perda de voltagem é causada pelos sobrepotenciais, que exigem potenciais de carga mais altos e diminuem o potencial de saída. Com base nisso, propomos um método de abordagem para investigar as variáveis relacionadas a esses problemas e sugerir estratégias de mitigação a serem testadas em sistemas reais. Além dos problemas relacionados às VRFBs, também investigamos as consequências da escolha de diferentes espécies ativas no desempenho de RFBs. Com esta abordagem nós: (i) mostramos como a geometria influencia na mitigação dos sobrepotenciais ôhmico e de concentração e como os parâmetros geométricos interagem com as condições operacionais; (ii) identificamos as condições operacionais que afetam a perda de capacidade e propomos uma estratégia de mitigação baseada na transferência de volume entre tanques no sentido inverso da contaminação cruzada líquida; e (iii) fornecermos insights úteis para entender como a escolha de espécies ativas em RFBs é relevante para a eficiência energética. Desta forma, conseguimos fornecer um embasamento teórico profundo para os experimentalistas entenderem os efeitos de diversas variáveis no desempenho de baterias de fluxo redox.

Abstract

PERFORMANCE OPTIMIZATION IN REDOX FLOW BATTERIES: A COMPUTATIONAL APPROACH

Redox flow batteries (RFBs) emerge as an alternative technology for the storage of energy on a large scale. The main characteristics of this type of device are related to its unique design, which provides scalability, decoupling between power and energy density, and the use of several types of active species. The more developed RFB to date is the all-vanadium redox flow battery (VRFB), which uses V^{2+}/V^{3+} in one half-cell and VO^{2+}(V(IV))/VO_2^{+}/(V(V)) in the other. Despite the advantages of this type of device, the VRFB still faces some drawbacks related to performance, which hinder its marketing penetration. Two of the most relevant problems are capacity loss and voltage loss. The capacity loss is caused by the non-ideal selectivity of the membrane, which allows the cross-contamination between the half-cells, leading to self-discharge reactions. The voltage loss is caused by overpotential that requires higher charging voltages and decreases the output voltage. Based on these problems, we propose a new computational method of approach to investigate the variables related to these problems and to suggest mitigation strategies to be further tested in real systems. Beyond the problems related to VRFBs, we also investigate the consequences of choosing different actives species in the performance of RFBs, based on this same method of approach. With this approach, we: (i) show how geometry influences the mitigation of overpotential and how geometric parameters interact with operating conditions; (ii) identify the operating conditions that affect the capacity loss and purpose a mitigation strategy based on the volume transfer between tanks in the reverse direction of net cross-contamination; and (iii) provide useful insights to understand how the choice of active species in RFBs is relevant to energy efficiency. Thus, we provide a set of theoretical backgrounds for experimentalists to understand the effects of several variables in the performance of redox flow batteries.

Contents

1 INTRODUCTION

This chapter provides an overview of the background: the storage technologies, the redox flow batteries, the issues with capacity loss and voltage loss, and the methods used (finite element method and chemometric analysis).

The increasing global demand for energy and the concerns about the environmental impacts of current energy production are driving forces for the improvement of renewable energy sources, such as solar and wind. However, this type of energy source suffers from intermittency, which is the major impediment to reliable electricity generation[1]. The intermittency leads to (i) the mismatch between electricity supply and demand, and (ii) complicates the direct transport to the power grid[2]. For this reason, the wide application of this type of energy source is highly dependent on energy storage systems (ESSs).

The principle of energy storage is transforming electrical energy into chemical, electrochemical, or mechanical energy. This allows the decoupling between the electricity generator from the electricity user[1]. In this situation, the power generated by off-peak time may be stored and then smartly delivered when the demand grows[1]. However, the requirements for energy storage are wide, including fast response time, high energy density, wide operating temperature, low cost, high round-trip efficiency, high lifetime, low rate of self-discharge, safety, and low environmental effects[3]. No known ESS meets all the requirements for large-scale energy storage. Despite this, some devices are already widely used, as pumped-hydro energy storage, and others are at demonstration levels, such as some advanced electrochemical capacitors and batteries[1].

In the first section, a brief review of the most important ESSs is shown. In section 1.2, the focus is the emergent redox flow batteries (RFBs), which are electrochemical energy storage devices based on the circulation of liquid electrolytes through porous electrodes, where the energy conversion electrical/electrochemical occurs. Sections 1.3 and 1.4 the drawbacks related to performance – capacity loss and voltage loss - are widely discussed. In section 1.5, the experimental evaluation of RFBs is discussed aiming at the understanding of the theoretical choices in the next chapters. Finally, in sections 1.6 and 1.7, the methods used in this thesis – finite element method (FEM) and chemometric analysis – are discussed.

1.1 Large-Scale Energy Storage

The ESSs are classified according to how energy is stored. **Figure 1** shows some examples based on this classification.

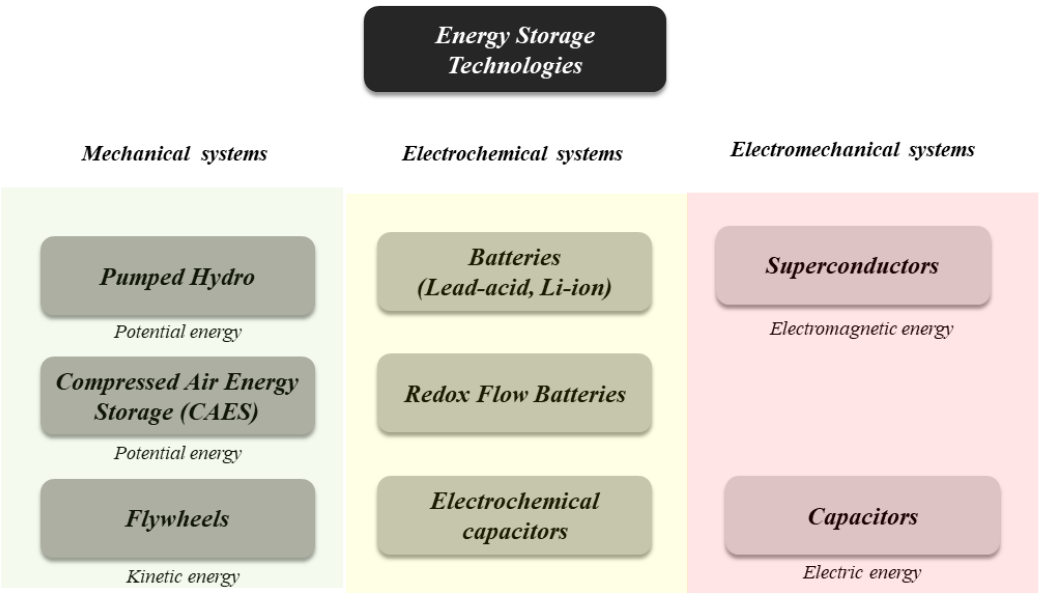


Figure 1. *Classification of ESSs based on the form of stored energy. Based on references*[2,4].

The pumped hydro storage is a very deployed technology and corresponds to most of the total installed capacity connected with the grid (96% in 2018 in the world[5]). The major advantages of this technology are the long working life, high capacity, and low cost[2].

The working principle of pumped hydro reminds the generation of energy by hydroelectric sources. During low-demand periods, the water is pumped up to a reservoir and the electric energy is converted into potential energy. When the demand grows, the water flow drives the turbines to generate electric power. Thus, this type of technology is restricted by geographical conditions and topography[2].

Another type of ESS based on potential energy storage is compressed air energy storage (CAES). The working principle of CAES is based on the compression of air using electrical energy. When demands grow, the air under high pressure is delivered to drive the turbines and generate electric power[6]. This type of storage technology shows high capacity, low cost, and long cycle periods of energy storage[2]. However, CAES suffers

from low energy efficiency due to changes in temperature caused by compression/expansion of the gas[2]. CAES plants also require suitable geological conditions, since the air compressed is stored in underground caverns.

The flywheels storage technology is based on the conversion of electric energy into kinetic energy by the angular speed of flywheels[4]. During peak-off times, an electromotor drives the flywheel to high angular velocities; thus, electric energy is converted into kinetic energy. When electric energy is required, the kinetic energy is reconverted to electricity and the angular velocities decrease. The main advantages of this technology are fast response, high power density, long cycle life, and environmentally friendly operation[1]. The disadvantages include high capital cost and high self-discharge during idling (capacity loss by friction)[7].

The main advantage of electrochemical storage devices is the no geographical requirement for installation. Beyond that, the electrochemical storage devices show standalone modular design and fast response[8]. This class is usually divided into batteries (e.g., lithium-ion, lead-acid), flow batteries (e.g., vanadium redox flow batteries), and electrochemical capacitors. In the first two devices, the energy conversion is based on the faradaic processes (redox chemical reactions), whereas the last one is based on the electric field between the electrode and the electrolyte.

One of the most developed electrochemical energy storage devices is the lithium-ion battery, widely employed in portable electronic devices, such as smartphones and laptops. The success of the lithium-ion battery is related to the high energy and power density, cost-effectiveness, high energy efficiency, and long operational life[9]. The low redox potential of lithium (-3.04 vs SHE) and the low molar mass (6.94 g/mol) are what make this chemical element so attractive, and allow the development of devices with high power and energy densities.

The working principle of lithium-ion batteries is the ion lithium transfer between the cathode and the anode. The anode is usually made of layered graphite and the cathode is usually made of layered lithium metal oxide (e.g., $LiCoO_2$, $LiFePO_4$)[10]. The layered structure of both anode and cathode materials allows the intercalation and deintercalation of Li^+, at the same time that current flows in the circuit. During discharging procedure, the external power source drives the oxidation in the anode and deintercalation of lithium:

$$LiC_6 \rightleftharpoons C_6 + Li^+ + e^- \ (anode) \qquad \textbf{(1)}$$

The current flows by the external circuit and the Li^+ is enabled to flow across a non-aqueous electrolyte, where a soluble lithium salt (e.g., $LiPF_6$) is dissolved and embedded in a separator felt. Thus, the electrons will drive the reduction in the cathode and the consecutive lithium intercalation:

$$Li^+ + e^- + MO_2 \rightleftharpoons LiMO_2 \text{ (cathode)} \quad \textbf{(2)}$$

Where *M* is a cation of metallic element, as cobalt.

Thus, the overall and general cell reaction is:

$$Li_xC_y + Li_{(1-x)}MO_2 \rightleftharpoons yC + LiMO_2 \quad \textbf{(3)}$$

Despite the advantages of lithium-ion batteries, the application in large-scale energy storage requires better safety, lower cost, wide temperature operational range, materials availability and higher capacities than this device can currently provide[1,10]. The safety issue is related to the possibility of explosion when the battery is short-circuited, overcharged, or overheated[2].

Because of the drawbacks of lithium-ion batteries, several other batteries have been studied, such as sodium-sulfur, sodium-nickel, and redox flow batteries. The main advantages of redox flow batteries are related to their modular design[11], which provides attractive scalability and flexibility for stationary energy storage. The flexibility is associated with the decoupling between power and energy capacity. The output power is controlled by the size of the stack, whereas the energy capacity is controlled by the volume of the electrolyte reservoirs.

1.2 Redox Flow Batteries

A typical design of a redox flow battery (RFBs) includes the anode and cathode compartments, the separator, two current collectors, two electrolyte reservoirs, and peristaltic pumps (see **Figure 2**).

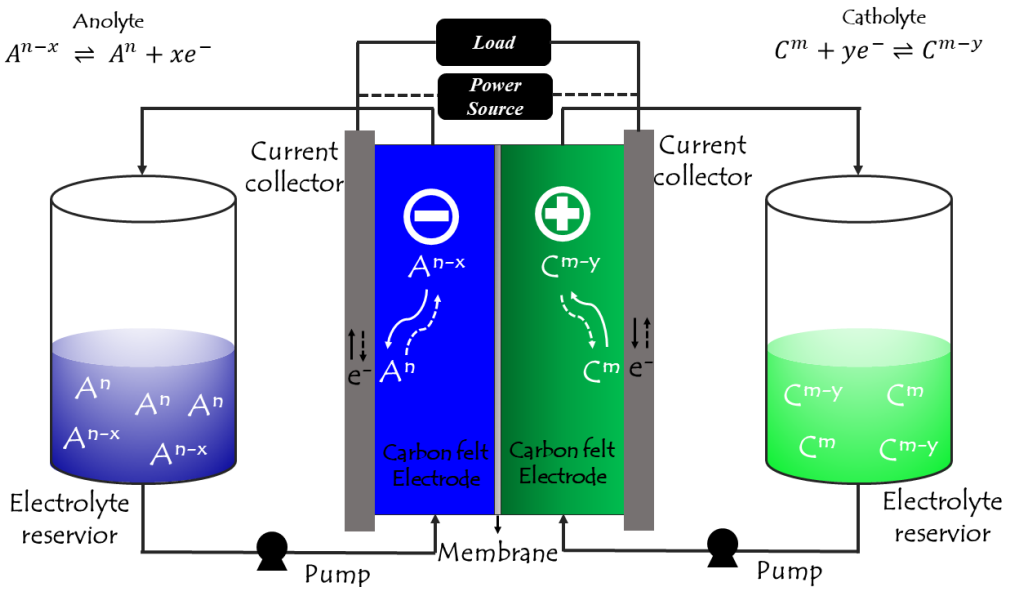


Figure 2. *Schematic representation and working principle of a redox flow battery.*

The energy conversion is based on the reversible electrochemical reaction of two redox couples dissolved in the electrolytes, the anolyte (A) in the anode and the catholyte (C) in the cathode:

$$yA^{n-x} + xC^m \rightleftharpoons yA^n + xC^{m-y} \qquad \textbf{(4)}$$

Where *x* is the number of electrons transferred in the anode reaction, *y* is the number of electrons transferred in the cathode reaction, *n* and *m* are the redox state of the oxidized anolyte and catholyte, respectively.

Both anode and cathode compartments are filled by porous electrodes (e.g., carbon felt), where electrochemical reactions take place. Both electrolytes flow continuously from reservoirs through the porous electrodes. The separator (e.g., ion-exchange membrane) allows the ionic flux between the half-cells and prevents the mix of electrolytes. And the current collectors are used to electrically connect the half-cells to the load/source and allow the external electron flow.

The first RFB was developed in the 1970s by Lawrence Thaller at the U.S. National Aeronautics and Space Administration (NASA)[12]. This device employed Cr^{3+}/Cr^{2+} as anolyte and Fe^{3+}/Fe^{2+} as catholyte. The main issue with the first RFB was the cross-contamination of active species between the half-cells and the consequent irreversible* capacity loss. Other problems in this type of device include low coulombic efficiency and

*In this case, the terms reversible and irreversible do not have thermodynamic meaning. This nomenclature is used just to determine if the capacity loss can be mitigated by concentration rebalancing (reversible) or not (irreversible).

irreversible capacity loss caused by hydrogen evolution due to the low redox potential of the Cr^{3+}/Cr^{2+} couple[2].

The first successful strategy for mitigation of cross-contamination in RFBs was achieved by Skyllas-Kazacos in 1988[13] by using two redox couples from vanadium: V^{2+}/V^{3+} as anolyte and VO^{2+}/VO_2^{+} as catholyte. This battery is called all-vanadium redox flow battery (VRFB). Although cross-contamination is still a problem in VRFBs, the products of the reactions between the two half-cells are the vanadium species themselves. Thus, the capacity loss in VRFBs is reversible and is mitigated by concentration rebalancing[14]. The cross-contamination in VRFBs is discussed in more detail in section 1.4.

The VRFBs are the most well-established redox flow batteries to date and drive the development of their entire class. However, this device suffers from some issues: (i) as the narrow working temperature window (between 10 and 40 °C)[8], which leads to active species precipitation; (ii) oxygen[15] and hydrogen[16,17] side reactions that lower cell efficiency and contributes to capacity loss; (iii) high cost of vanadium[18] and the ion-exchange membrane[19]; (iv) low energy density[20]; and (v) the reversible capacity loss[14].

To overcome some of these issues, the search and development of new redox-active materials increased since 2010[21]. Usually, the systems are divided into aqueous or nonaqueous. The benefits of using aqueous systems are the negligible environmental impact, the low cost, and the high conductivity provided by high soluble salts in water. This allows the operation of the cell in high currents without larger voltage losses[22]. However, the most problematic aspect of using an aqueous medium is the narrow electrochemical window of water, which restricted the use of species with absolute high redox potentials[23]. The hydrogen and oxygen evolution (i) lowers efficiency, (ii)limits power density, and (iii) increases capacity loss due to the use of a certain percentage of current to drive the parasitic reactions[24].

Figure 3 illustrates the challenges faced by the RFB's market penetration. These challenges are divided according to the components of RFB.

Active Species. Regarding the classification of active species, they can be metal (e.g., vanadium, iron, cerium), metal-containing organic (e.g., metallocene[25,26], metallic complexes[27,28]), and metal-free organic. This last class is also called redox organic molecules (ROMs). The choice of a suitable redox active material is usually related to the theoretical energy density:

$$E = nFc_{act}U \qquad \textbf{(5)}$$

Where n is the number of electrons transferred in the reactions, F is the Faraday's constant, c_{act} is the concentration of the active species and U is the theoretical cell redox potential.

As can be seen in ***Equation 5***, two properties are related to the energy density: the solubility and the redox potential. In the case of ROMs, these properties can be tunable and this drives the attention of researchers to the development of organic materials for RFBs application. ROMs also have benefits related to low cost and sustainability, since these materials are composed of earth-abundant elements[29].

Despite solubility and redox potential being the target properties for choosing the suitable active species, their consequences on the battery's efficiency are not known yet.

The most relevant drawback of using organic materials as active species in RFBs is the instability of electrochemically generated radicals. The reduction or oxidation of organic compounds usually occurs with coupled chemical reactions, such as dimerization or cleavage. The formation of unstable radicals is very common[30,31] and structural modification plays a significant role in the improvement of chemical persistence of electrochemically generated radicals. The strategies for mitigating the rate of coupled chemical processes are wide, but usually involve the increase of spin delocalization[32,33] and steric protection of unpaired electron[30].

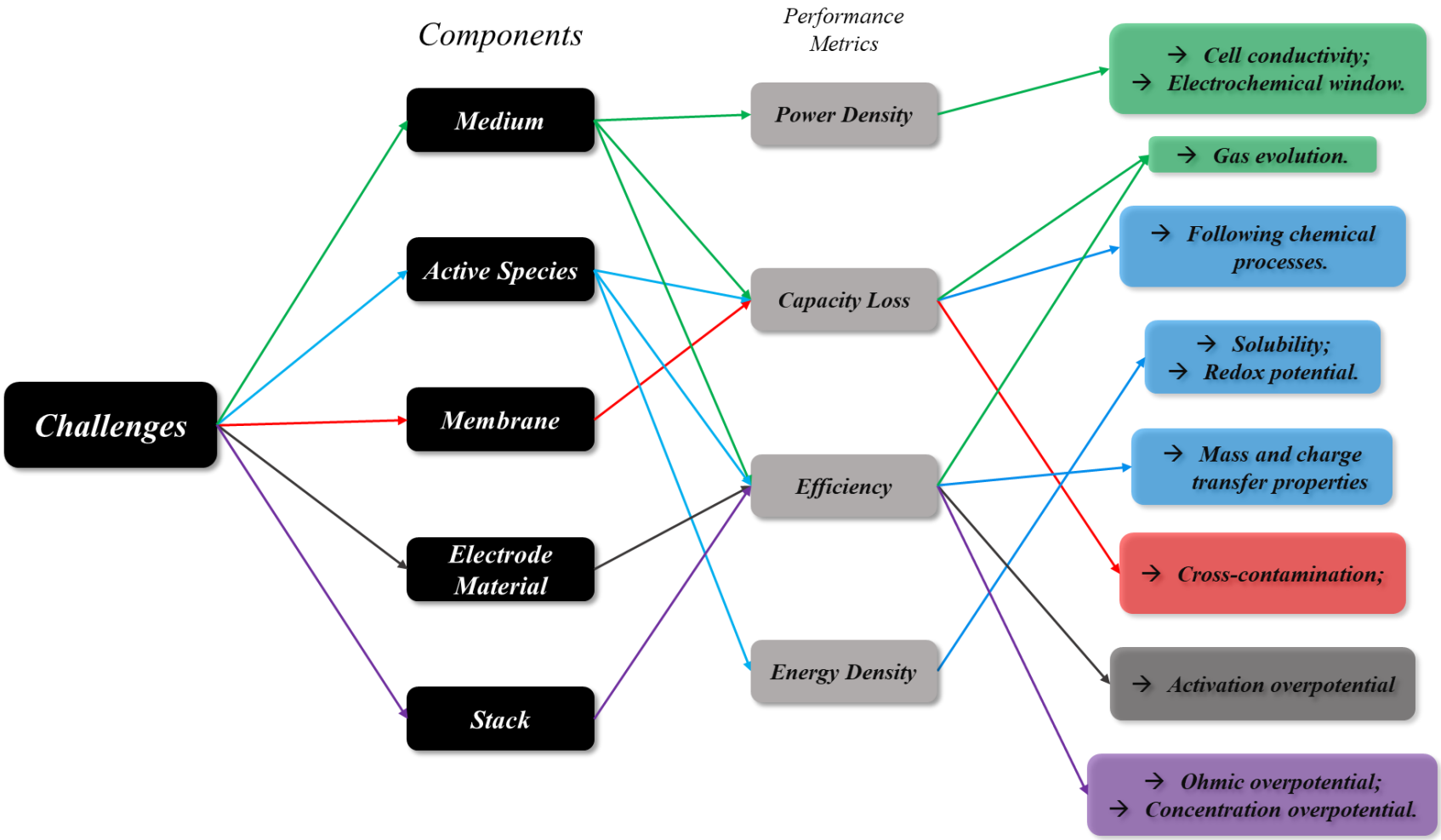


Figure 3. *Challenges of redox flow batteries per cell components.*

Other key properties controlling RFB performance are the mass transport to the electrode and the electron transfer rate[34]. Despite the convective flow in the cell, the diffusional transport to the electrode plays a role in the effective mass transfer from bulk to the surface of the electrode due to the existence of a convective-dependent diffusional layer[35,36]. Thus, the magnitude of diffusion coefficients affects the concentration overpotential and consequently the voltage efficiency. The electron transfer rate is even a more important property because determines the usability of the cell to deliver a desirable current density with high efficiency[34].

Electrode material. The development of electrodes is also very important to achieve competitive RFBs. The development of electrodes is especially important if the kinetics of active species is sluggish, as in VRFBs. Many researchers have studied methods to improve the electrochemical activity of electrodes to increase the reversibility of vanadium reaction aiming for a better energy efficiency[37].

Membrane. The membrane affects the capacity loss due to its selectivity but also plays an important role in the cost-effectiveness of RFBs[38]. One of the most important drawbacks in VRFBs is the cross-contamination caused by the non-selective transport of ions across the Nafion membrane. The cross-contamination causes the self-discharge of the cell and the consequent imbalance of active species between the half-cells[14], which leads to reversible capacity loss. Despite being reversible, the capacity loss adds cost to the VRFB operation.

Stack. The performance of RFBs is also dependent on stack design. A RFB can operate with two designs based on the electrode configuration: flow-through and flow-by[39] (see **Figure 4**). In the flow-through design, the entire electrolyte is forced through the porous electrode, whereas in the flow-by design the electrolyte is pumped by the electrode with specified flow channels defined by flow fields, as serpentine flow field (SFF) and interdigitated flow field (IFF)[39].

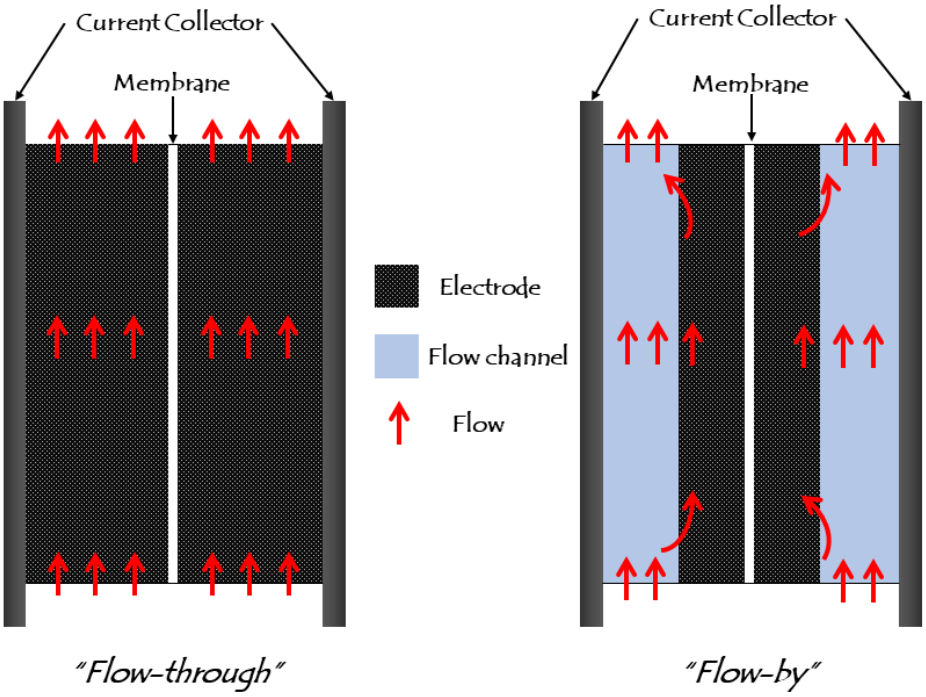


***Figure 4**. Representative scheme of stack designs: flow-through and flow-by.*

The flow-through design demands thicker electrodes than the flow-by design. This is because the pump power required for electrolyte circulation is large in the flow-through design, and the mitigation strategy is the enlargement of the electrode. However, this strategy led to a typical large ohmic overpotential[40]. The main advantage of the flow-through design is the uniformity of active species distribution, which is related to the magnitude of concentration overpotential. However, the choice of the best design depends on several factors: scale, cell dimensions, flow velocity, electrode compression, etc. Indeed, there are no decisive studies about which design leads to the best performance[41].

1.3 Strategies for Efficiency Improvement in VRFBs

Energy efficiency is defined as the relation between the retained energy in the discharge procedure by the applied energy in the charging procedure. The energy can be obtained by integration over time of the product between current and cell potential[42]:

$$Energy\ Efficiency\ (EE) = \frac{\int_0^{t_d} I_d(t)V_d(t)dt}{\int_0^{t_c} I_c(t)V_c(t)dt} \times 100\% \qquad \textbf{(6)}$$

Where I_d is the current in the discharge procedure, V_d is the discharging potential, I_c is the current in the charging procedure, V_c is the charging potential, t_d is the discharging time and t_c is the charging time.

The integration over time of the current is equal to the delivered charge in the discharging procedure (q_d) and the retained charge in the charging procedure (q_c). This quotient is equal to the coulombic efficiency:

$$Coloumbic\ Efficiency\ (CE) = \frac{\int_0^{t_d} I_d(t)dt}{\int_0^{t_c} I_c(t)dt} \times 100\% = \frac{q_d}{q_c} \times 100\% \qquad \textbf{(7)}$$

The CE is reduced by the crossover of active species across the membrane[43], and side reactions[44]. For instance, the oxygen evolution reaction (OER) and hydrogen evolution reaction (HER) consume a part of the current directed to the active species[24] and lead to the inactivation of active sites due to the bubble formation[45].

The quotient between the mean discharging potential and the mean charging potential at constant current is defined as voltage efficiency (VE)[22]:

$$Voltage\ Efficiency\ (VE) = \frac{\frac{\int_0^{t_d} V_d(t)dt}{t_d}}{\frac{\int_0^{t_c} V_c(t)dt}{t_c}} \times 100\% \qquad \textbf{(8)}$$

The VE is reduced by the overpotential (ohmic, concentration, and activation) generation. In the discharging procedure, the overpotential reduces the potential delivered by the cell. In the charging procedure, the overpotential must be overcome and higher potentials must be applied to charge the cell (see **Figure 5**). In the absence of overpotential, just the thermodynamic potential is observed (open circuit potential, OCP) and the voltage efficiency is equal to 100%.

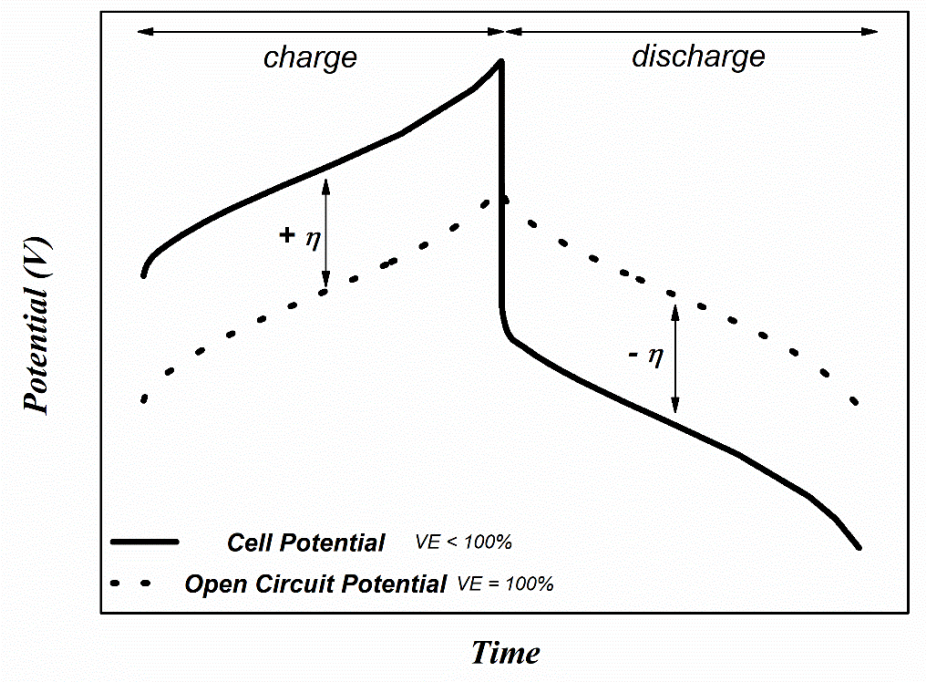


Figure 5. *Cell potential and open circuit potential for a typical redox flow battery.*

The overpotential is usually divided into ohmic, concentration, and activation. The ohmic overpotential is related to the overall electric resistance of the cell. The concentration overpotential is related to the mass transfer resistance of active species from the bulk to the electrode surface[46]. And the activation overpotential is related to the potential needed to overcome the activation energy of the electrochemical reaction[46]. These three overpotentials can be analyzed separately and each one can be mitigated using different strategies, from geometric modification of electrodes to the choice of active species, membranes, and electrodes.

Thus, the energy efficiency can be redefined in a more intelligible way as the product of coulombic and voltage efficiencies:

$$EE = CE \times EE \qquad \textbf{(9)}$$

Another typical efficiency definition for redox flow batteries includes the pumping energy consumption and is defined as the battery or system efficiency:

$$Battery\ efficiency\ (BE) = \frac{\int_0^{t_d} I_d(t)E_d(t)dt - \int_0^{t_d} P_{pump}dt}{\int_0^{t_c} I_c(t)E_c(t)dt - \int_0^{t_c} P_{pump}dt} \times 100\% \qquad \textbf{(10)}$$

Where P_{pump} is the pump power.

The magnitude of the pump power is related to how compressed is the carbon felt, the design of the cell (flow-through and flow-by), the electrode thickness, and the length of the residence path.

Cross-contamination. The key property to avoid cross-contamination is the membrane selectivity. The use of perfluorinated sulfonic acid membranes, such as Nafion, is the most used membrane for VRFBs due to its high proton conductivity and chemical stability[38]. The drawbacks of this kind of separator include the high vanadium permeability and high cost[47].

The cross-contamination in the VRFBs leads to self-discharge reactions. In the negative electrode[48]:

$$VO^{2+} + V^{2+} + 2H^{+} \rightarrow 2V^{3+} + H_2O \quad \textbf{(11)}$$

$$VO_2^{+} + 2V^{2+} + 4H^{+} \rightarrow 3V^{3+} + 2H_2O \quad \textbf{(12)}$$

In the positive electrode[48]:

$$V^{2+} + 2VO_2^{+} + 2H^{+} \rightarrow 3VO^{2+} + H_2O \quad \textbf{(13)}$$

$$V^{3+} + VO_2^{+} \rightarrow 2VO^{2+} \quad \textbf{(14)}$$

The self-discharge reactions lead to the imbalance of active species between the half-cells due to the different transfer properties among the four redox states of vanadium[43]. As the concentration of active species varies with time in each half-cell, the applied charge in the charging procedure will be greater than the charge delivered in the discharge one, leading to CE < 100%.

To mitigate this issue, several authors have proposed alternative non-perfluorinated membranes[38], anion exchange membranes[49], and porous membranes[50] for VRFB application. Beyond selectivity, several other properties must be considered when choosing the suitable membrane: ionic conductivity, working current density, preparation technologies, stability, and cost[38]. The anion exchange membranes show lower permeability, but have low stability and low conductivity[38]. The porous membranes are

stable and cheap but are inefficient in the ion exchange process, leading to unsatisfactory performance[38].

OER and HER. As aforementioned, the OER and HER are responsible for lowering the expected 100% of CE. The practical strategies to mitigate these reactions are: to limit the operational SoC window between 10 and 90%[43]; choose properly the electrode material[17]; and use inhibitors[51] (e.g., bismuth).

Activation overpotential. The mitigation of activation overpotential occurs on two fronts: (1) the choice of active species with intrinsic fast electron transfer rate and (2) the development of electrocatalysts with high electrochemical performance.

The two electron transfer processes involved in VRFBs are sluggish[52]:

$$V^{2+} \rightleftharpoons V^{3+} + e^{-}, k^{0}(glassy\ carbon) = 5.4 \times 10^{-5}\ cm\ s^{-1} \quad \textbf{(15)}$$

$$VO_2{}^{2+} + 2H^{+} + e^{-} \rightleftharpoons VO^{2+} + H_2O, k^{0}(glassy\ carbon) = 1.3 \times 10^{-5} cm\ s^{-1} \quad \textbf{(16)}$$

The sluggish rate of the above reactions lead to large activation overpotential and lower energy efficiency, especially when large operational currents are demanded. Because of this, recent authors have studied different modifications in the carbon-based electrodes to improve the electron transfer rate of vanadium reactions[53]. The electrode modification is also important to increase the chemical stability, decrease ohmic resistance and prevent gas evolution reactions[53].

Concentration overpotential. The concentration overpotential directs the greatest attention of researchers in the issue of efficiency improvement for VRFBs. Two types of mitigation strategies are widely studied: (1) dynamic control of the flow rate[54–61] and (2) modification of cell design. All these strategies are based on increasing the uniformity of concentration.

Unlike a constant flow rate, the variation in flow rate throughout the operation of the battery allows the lowing of pumping energy consumption during the intermediate SoCs (lower rate), whereas significantly decreasing the concentration overpotential in the extremes SoCs (higher rate)[56]. This type of strategy also leads to heat dissipation, inhibiting the possible precipitation of active species[56].

The modification of cell design includes (i) the use of optimized flow fields to improve the distribution of active species in flow-by designs[62–65] and (ii) the proposal of new electrode geometries[66–68] that allow a better mass distribution in the outlet channels, where the concentration overpotential is usually larger.

Ohmic overpotential and pumping energy consumption. The mitigation strategies for ohmic overpotential also affect the pumping energy consumption. Two strategies are identified: (1) compression of the electrode and (2) the use of thin electrodes.

The compression of the electrode decreases the porosity of the electrode, which defines the fraction of electrolyte inside each half-cell. As the electrode is more conductor than the electrolyte, this operation decreases the cell electrical resistance:

$$\eta_o = i_{appl} \frac{L_{electrode}}{\varepsilon^{3/2}\sigma_l^e} \qquad \textbf{(17)}$$

Where $L_{electrode}$ is the electrode thickness, i_{appl} is the operational current density, ε is the porosity of the electrode, σ_l^e is the ionic conductivity of the electrolyte.

Usually, the compression of the electrode is made *in situ,* leading to the narrowing of the electrode. These two processes - compression and narrowing of the electrode -mitigate the ohmic overpotential. However, this same operation difficult the electrolyte circulation requiring larger pump power and consequent increase in the pumping energy consumption.

Park et al.[69] have shown the improvement of voltage efficiency caused by the compression of the electrod*e in situ*. However, the coulombic efficiency is highly affected when a porosity of 0.873 is achieved. This may be related to the increase of pressure inside the cells, which increases cross-contamination.

1.4 Capacity loss in VRFBs

The theoretical capacity is defined as a measure of the amount of charge stored in the cell[22]:

$$C = nFc_{act} \quad \textbf{(18)}$$

Where n is the number of electrons transferred, F is the Faraday's constant, and c_{act} is the concentration of active species.

This capacity varies over cycles due to the existence of side reactions and cross-contamination. This phenomenon is named capacity loss and is responsible for the decrease of cell lifespan.

The OER takes place in the positive electrode and the HER takes place in the negative electrode. As these two electrochemical reactions have distinct kinetics and thermodynamics, they occur in an unbalanced way in the half-cells, passivating the active sites and consuming the current non-uniformly in each half-cell. This leads to the imbalance of active species in the cell, which corresponds to a limiting half-cell reaction and consequent capacity loss[45].

However, the most relevant issue for capacity loss in VRFBs is cross-contamination. Cross-contamination is driven by four types of transport: diffusion, migration, osmosis, and electroosmosis (see **Figure 6**).

Diffusion. The diffusion occurs due to the concentration difference of vanadium ions between the positive and negative half-cells[43] and is the main transport mode in the cross-contamination[70,71].

The imbalance of active species drives by diffusion depends on the diffusion coefficients in membrane[44], which are large for V^{2+} and V^{3+} in the negative electrode than for VO^{2+} and VO_2^+ in the positive electrode. Because of this, the total concentration of vanadium ions increases in the positive half-cell and decreases in the negative half-cell. Thus, the self-discharge reactions in the positive electrode occur more frequently and the negative electrode limits the cycle operation. The diffusion direction is independent of the direction of the electric field.

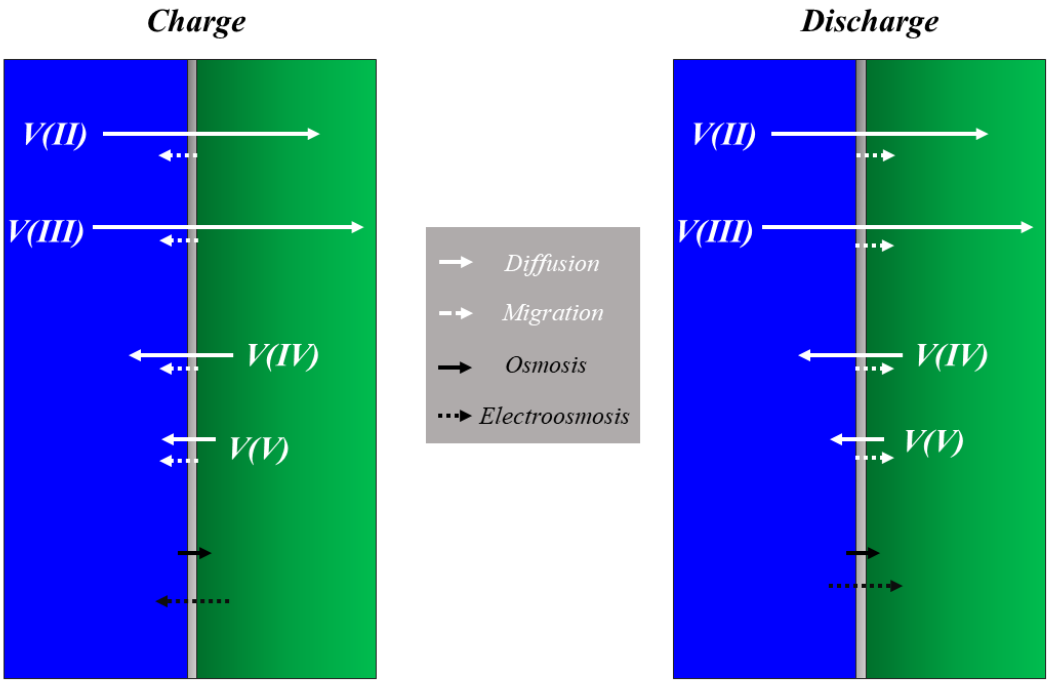


Figure 6*.The direction of vanadium flux due to diffusion, migration, osmosis, and electroosmosis in the charging and discharging procedure.*

Migration. Migration across the membrane is driven by the potential difference between the positive and negative half-cells. The migration direction depends on the direction of the electric field. In the charging procedure, the conventional electric current flows from the negative to the positive electrode; thus, the generated ionic potential drives the flow of positively charged species from the positive to the negative electrode. In the discharging procedure, the direction is reverted.

The migration has a minimal effect on the net vanadium crossover[71]. First, because of the change of direction between the charging and discharging procedure. Second, because of the similar magnitude of ionic conductivity among the vanadium species[44].

Osmosis. Osmosis is a convective transport that occurs, in this case, due to pressure differences between the half-cells. This pressure difference arises from the difference of viscosity between the positive and negative electrolyte[72].

The viscosities of electrolytes depend on the concentration of active species, concentration of supporting electrolyte, temperature, and state of charge (SoC)[72,73]. Usually, the negative electrolyte is more viscous than the positive electrolyte; thus, the pressure needed to flow the negative electrolyte is greater. This larger pressure in the

negative electrode drives the solution to flow across the membrane from the negative to the positive half-cells.

Electroosmosis. Electroosmosis is a convective transport that occurs due to the drag of the solvent caused by the motion of an excess of charge in an applied electric field.

Nafion membranes contain negatively charged groups (-SO_3H, sufonyl)[38]. When an electric field is applied to the membrane, the cations move to the negative side, and the anions move to the positive side. However, the number of cations in the membrane is larger because of the existence of fixed sulfonyl groups. Thus, the motion of this excess of charge drags the solvent[36] (see **Figure 7**).

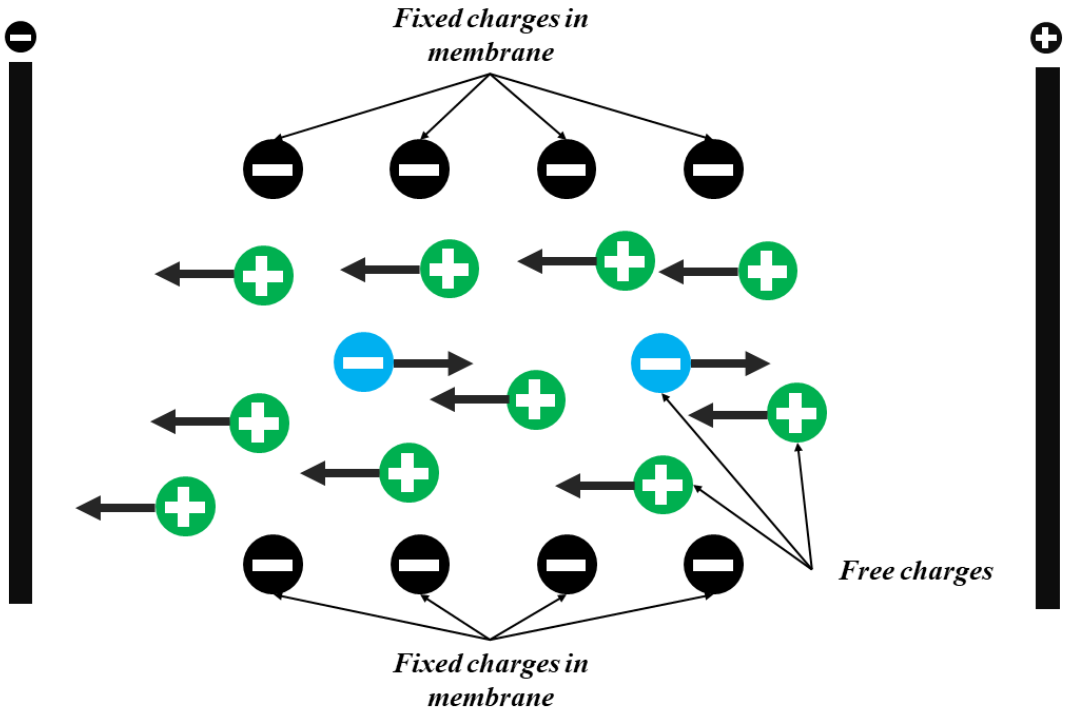


Figure 7. *Representation of electroosmosis flow in a Nafion membrane.*

The direction of electroosmosis flow depends on the direction of the electric field. In the charging procedure, the potential difference drives the flow from the positive to the negative electrode[70]. In discharging procedure, the flow is reverted.

The electroosmosis flow is an important mode of transport because affects the magnitude and the direction of vanadium crossover. In s-Radel membranes, the electroosmosis is even more important than the diffusion[71].

Water transfer. The convection across the membrane caused by osmosis and electroosmosis causes the water transfer from one half-cell to another. This process increases the capacity loss of the cell because dilutes one electrolyte and concentrates the other[45,72]. The self-discharge reactions (**Equations 11-14**) also are sources of water transfer between the half-cells[74].

Mitigation strategies. The strategies used to mitigate the capacity loss in VRFBs are focused in minimize the driving forces discussed above.

In the discharging procedure, the direction of migration and convection amplifies the disbalance caused by diffusion (see **Figure 6**), whereas in the charging procedure the direction of electroosmosis and migration is the reverse of net diffusion. A mitigation strategy consists in increasing the electroosmosis flow by using higher applied current densities in charging than in discharging[75]. As mentioned above, electroosmosis depends on the electric field. Increasing the current will turn the electroosmosis flow superior to the diffusional flow only in the charging procedure.

Another mitigation strategy is increasing the diffusional flow of vanadium ions from the positive to the negative half-cell to offset the intrinsic higher diffusion coefficient of V^{2+} and V^{3+}. This can be done by employing a larger concentration of active species in the positive electrode than in the negative electrode[76]. A good result was obtained by Lu et al.[76] using 2 mol L^{-1} of active species in the positive electrode and 1.04 mol L^{-1} in the negative electrode. When the concentration of active species is equal, the discharge capacity is lowered by 20.5% after 46 cycles. With the aforementioned asymmetric concentrations, the capacity loss remained almost unchanged.

Instead of applying different concentrations, Park et al[77] have used asymmetric electrolyte volumes. The aim was to compensate in advance for the volume change caused by water transfer and the imbalance of active species.

Another type of migration strategy is automatic rebalancing, which consists of the volume transfer between the electrolyte in the reverse direction of net cross-contamination. This method is cost-effective, simple and it was tested in medium-scale batteries[78]. Wang et al.[79], were the first to propose this strategy in 2017. They tested the automatic rebalancing by placing a connector between the two electrolyte tanks for three different conditions.

For VFB-0 (Vanadium Flow Battery – case 0), there was no flow between the tanks. For VFB-1, VFB-2, and VFB-3 the volume transfer occurs when the positive electrolyte volume increases by 10, 20, and 30%, respectively.

In a regular operation, the electrolytes have the same volume before the charging procedure. After some cycles, the volume changes, increasing in the positive electrolyte and decreasing in the negative electrolyte. When a connector is placed between the tanks, the excess volume in the positive tank flows to the negative tank, leading to a partial volumetric rebalancing. But, the most important part of this process is the recuperation of vanadium species in the negative half-cell due to the self-discharge reactions (see again **Equations 11-14**).

For an applied current density of 80 mA cm^{-2}, the best results were achieved by VFB-3 case. Whereas, for an applied current density of 160 mA cm^{-2}, the best results were achieved for VFB-2 case.

Thus, the applied current density influences the best condition of volume transfer between the tanks. Despite, this being a low-cost, simple, and well-proved mitigation strategy, it can be improved. For example, it can be determined which more operating conditions (e.g., concentration, volumetric flow) influences the capacity loss and how we can calculate the optimum condition considering these effects.

1.5 Experimental evaluation of VRFBs

Experimental setup. To understand the simulation steps taken in this thesis it is necessary to first understand how experiments are carried out to evaluate the performance of VRFBs. **Figure 8** shows the components of an experimental VRFB in a flow-through design.

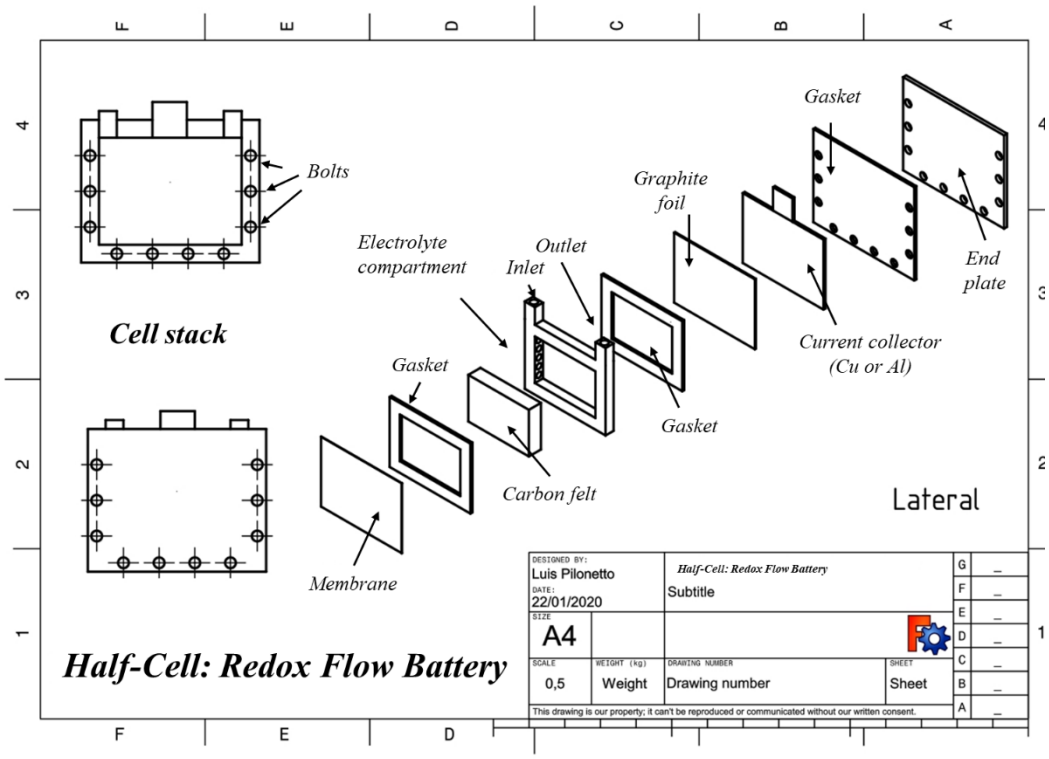


***Figure 8**. Components of a VRFB and cell stack. Based on references*[80,81].

The basic structure described in the scheme from **Figure 2** is also present in **Figure 8**: the electrodes of carbon felt, the electrolyte compartment, and the membrane. Rubber gasket seals and steel tie-bolts are used to compress the cell stack[82].

Since the electrolyte is oxidizing, no metallic component should be in contact with the electrolyte[82]. Because of this, chemically resistant polymers (e.g., PTFE) are used to produce the battery components. The only metallic component is the current collector of cupper or aluminum, usually separated from the electrolyte by graphite foils.

Performance evaluation. The first step in performance evaluation of a RFB consists in determining what charging currents can be applied and at what the initial voltage would be at those currents[83]. This is done by linear sweep voltammetry (LSV), as shown in **Figure 9**.

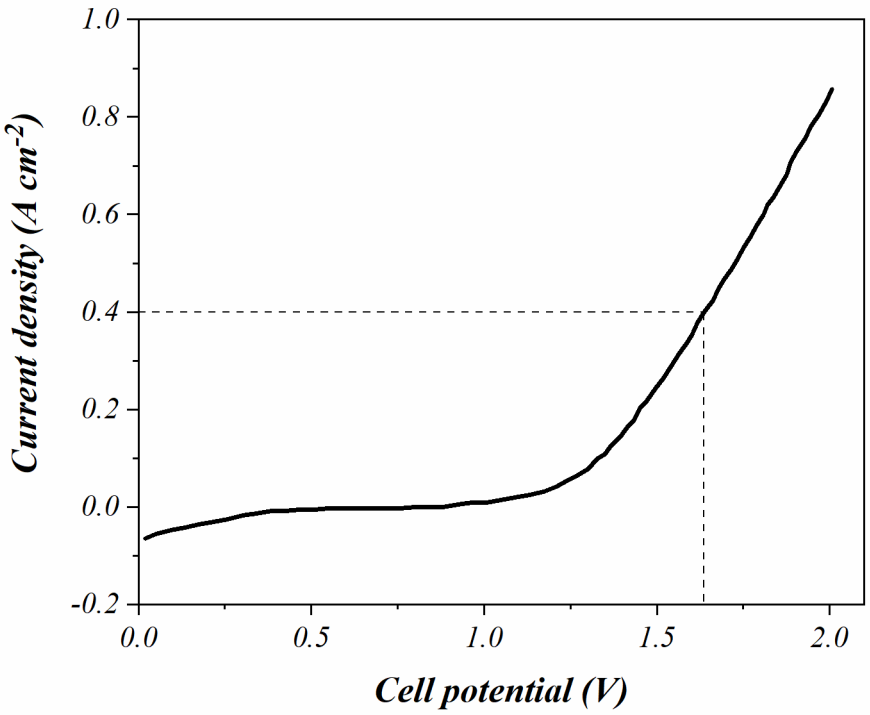


Figure 9. *A typical Linear Sweep Voltammetry for a VRFB. From reference* [83].

In this example, the battery operating in 0.4 A cm^{-2} (400 mA cm^{-2}) can deliver an output cell voltage of approximately 1.63 V.

The charging/discharging cycling is performed using a chronopotentiometry test (constant current) to charge and discharge the battery to obtain the cell voltage versus time as response[83]. This data allows the calculation of CE, VE, and EE as described in **Equations 7, 8,** and **9**, respectively. The RFB can be charged and discharged by two methods: voltage cut-off or time cut-off [83]. The voltage cut-off is usually used to fully charge and discharge the battery, whereas the time cut-off is usually used to discharge and charge the battery at a specific SoC[83]. **Figure 10** shows an example of the charging/discharging cycle for a VRFB operating at various current densities in the voltage cut-off method. In this case, the cell voltage *versus* time curve is replaced by cell voltage *versus* capacity, which is more common.

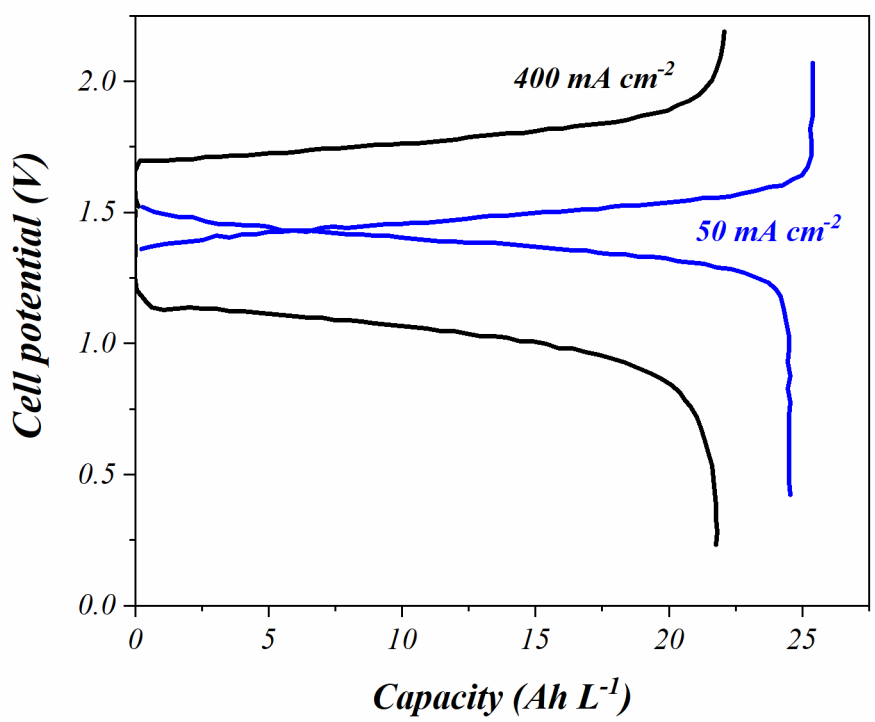


Figure 10. *A typical single charging/discharging cycle for a VRFB at different values of current densities. From reference* [83].

As the current density increases, the overpotential increases requiring higher charging voltages and delivering lower discharging voltages. The increase of applied current density also leads to a smaller use of total capacity due to the difficulty of maintaining the required current.

Cell cycling is also utilized to evaluate the capacity loss of the battery across cycles. The decrease of time discharge at the same voltage cut-off indicates this capacity loss, as can be seen in **Figure 11**. The shortening of the cycles indicates the limitation of discharging caused by the disbalancing of active species in the negative half-cell, as discussed in the previous section.

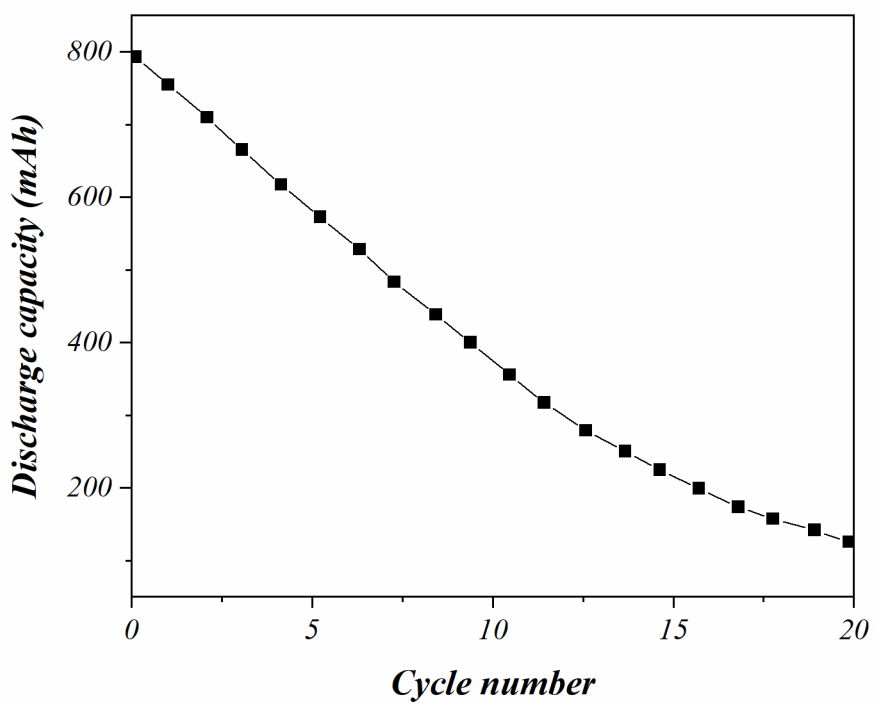


Figure 11. *Discharge capacity versus cycle number for a typical VRFB. Data from reference*[43].

1.6 Finite element method (FEM)

There are several problems for which we cannot obtain the analytical solution for the partial differential equations (PDEs). To deal with this, a numerical method can be used to approximate the exact solution. The first step of the finite element method is discretization, which consists of dividing the continuum medium of interest into well-defined components[84]. These components are called elements and the ends are referred to as the nodes (see **Figure 12**).

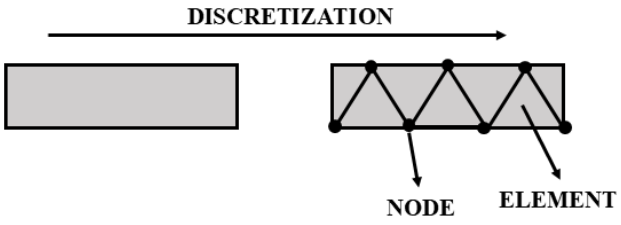


Figure 12. *Discretization of a continuum medium by finite element method.*

The next step is to approximate the dependent variable in the PDE (we called this variable *u*) by a guess (or trial) algebraic function u_h described by linear combinations of basis functions[85]. We use linear functions to illustrate this step, but other functions may be chosen (e.g. quadratic functions).

$$u \approx u_h = \sum_i u_i \psi_i \qquad \textbf{(19)}$$

Where ψ_i denotes the basis functions and u_i denotes the coefficients of the functions.

Figure 13 shows the real and approximated solutions of *u* along the *x*-axis. The linear basis functions have a value of 1 at their respective nodes and 0 at other nodes. This ensures the overlapping of functions along the *x*-axis[85].

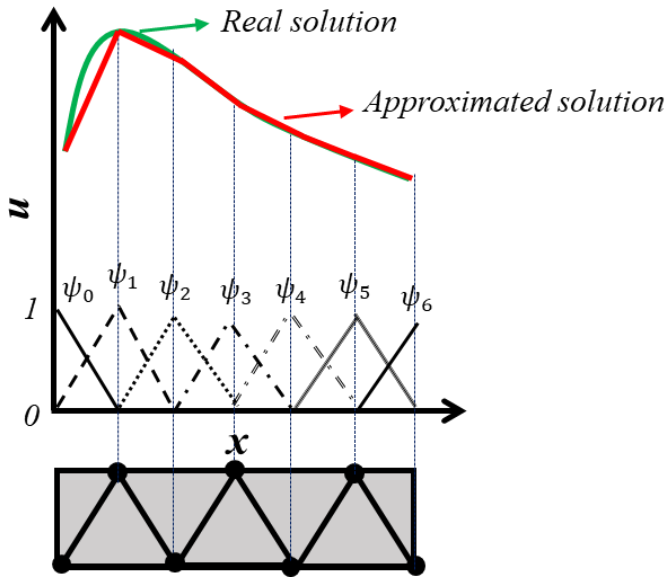


Figure 13. *Real and approximated solutions of variable u along the x-axis.*

The next step is determining the coefficients of these functions by minimizing the domain residual by mathematical techniques. The domain residual is defined as the difference between the exact solution and the approximated solution. For example, considering a simpler case (one-dimensional) based on the following governing differential equation[86]:

$$AE\frac{d^2u}{dx^2} + q_0 = 0 \qquad \textbf{(20)}$$

Where *A*, *E* and q_0 are constants.

This equation can assume the following approximate solution in a determined boundary condition, based on a second-order polynomial[86]:

$$u_h = u_2(x^2 - 2Lx) \quad \textbf{(21)}$$

Where u_2 is the coefficient.

Substituting **Equation 21** in **Equation 20**, we get [86]:

$$AE(2u_2) + q_0 = R_d \quad \textbf{(22)}$$

For the exact solution, the domain residual is equal to zero, as described in **Equation 21**. Setting $R_d = 0$, we can determine the coefficient u_2 for this case.

The General Weighted Residual (WR) method is based on the minimization of residual at all points in the domain. Considering an approximate solution of the form of **Equation 19**, the WR statement can be written as [86]:

$$\int_\Omega W_i R_d d\Omega = 0 \; for \; i = 1, 2, \dots, n \quad \textbf{(23)}$$

Where W_i are appropriately chosen weighting functions to minimize the residual over the entire domain (Ω) and R_d is the domain residual.

The choice of weighting functions (W_i) is arbitrary, but Galerkin suggested the idea of *W* being the same as the trial functions[86]. For example, for a situation where the domain residual is dependent on *x*, the residual cannot be set 0 everywhere in the domain[86]:

$$R_d = u_1 \left(\frac{\pi}{L}\right)^4 (EI) sin\,(\pi x/L) - q_0 \quad \textbf{(24)}$$

By the Galerkin procedure, we get[86]:

$$\int_0^L \boldsymbol{sin}\left(\frac{\boldsymbol{\pi x}}{\boldsymbol{L}}\right)\left[u_1 \left(\frac{\pi}{L}\right)^4 EIsin\left(\frac{\pi x}{L}\right) - q_0\right] dx = 0 \quad \textbf{(25)}$$

Where the first sinusoidal function (in bold) represents a weighting function and the second term is the domain residual. From this equation, we can get the value of the coefficient u_1.

This method requires the use of trial solutions that satisfy the boundary conditions of the problem[86].

Another technique is the Weak Form of the Weighted Residual Method, which reduces the continuity requirement on the trial functions[86]. This is done by applying the standard formula for integration by parts for a definite integral, that is:

$$\int_{\alpha}^{\beta} udv = [uv]_{\alpha}^{\beta} - \int_{\alpha}^{\beta} vdu = (uv)|_{\beta} - (uv)|_{\alpha} + \int_{\alpha}^{\beta} vdu \qquad \textbf{(26)}$$

Example:

For the following governing differential equation [86]:

$$AE\frac{d^2u}{dx^2} + ax = 0 \qquad \textbf{(27)}$$

Applying the weighted residual statement [86]:

$$\int_0^L W\left[AE\frac{d^2u_h}{dx^2} + ax\right]dx = 0 \qquad \textbf{(28)}$$

i.e.

$$\int_0^L WAE\frac{d^2u_h}{dx^2}dx + \int_0^L Waxdx = 0 \qquad \textbf{(29)}$$

Or

$$\int_0^L Wd\left(AE\frac{du_h}{dx}\right) + \int_0^L Waxdx = 0 \qquad \textbf{(30)}$$

For this case, $u = W$ and $v = AE\frac{du_h}{dx}$. Integrating the first term of the above equation, we get [86]:

$$\left[WAE\frac{du_h}{dx}\right]_0^L - \int_0^L \left(AE\frac{du_h}{dx}\right)\frac{dW}{dx} + \int_0^L Waxdx = 0 \qquad \textbf{(31)}$$

In the weighted residual statement, we had the term d^2u_h/dx^2 and, in the weak form, we get du_h/dx. Thus, the trial function should at least be quadratic for the original weighted residual statement, whereas in the weak form even linear trial functions can be used [86]. Therefore, the continuity demand on the trial function gone down and that on the weighting function increases [86]. This allows the use of a much wider of trial functions.

***Advantages of the FEM*[84–86].** The advantages of using the finite element method are the great freedom in the selection of discretization and then the use of complex geometries; the well-developed theory, which turns the method consolidated and allows the computation of total residual by summing up the residual of each element; and the coupling between physics by using software packages. When several physics are utilized in a same numerical model, we termed this model as multi-physical.

1.7 Chemometric analysis

Chemometrics consists of a wider quantity of techniques, from the exploratory analysis of chemical data (e.g., PCA, HCA) and classification algorithms (e.g., KNN, SIMCA) to the design of experiments. This last one is the focus of this work

The purposes of an experimental design are[87]: (i) screening the variables that are important for the interest response; (ii) the optimization of the response; (iii) saving time and; (iv) obtaining a quantitative model. To achieve these benefits of experimental design two approaches are used:

***Factorial design*[88].** The factorial design focus on the screening of variables. The purpose is to evaluate the effect of each variable on the response, classifying them as important or not to the response.

The number of variables dictates the number of experiments. If two variables are used, we have a 2^2 factorial design and, then, 4 experiments as shown in **Table 1**. The variables are represented as v_1 and v_2.

Table 1.*2^2 factorial design (codified).*

ID	v_1	v_2
1	*-1*	*-1*
2	*+1*	*-1*
3	*-1*	*+1*
4	*+1*	*+1*

When these experiments are run, we obtain the response of interest. For example, if v_1 is the temperature and v_2 is the type of catalyst, the response can be the yield, in percentage, of the reaction under study. From these values of variables and responses, the effect of each variable (**T**) can be calculated as[88]:

$$\boldsymbol{T} = (\bar{y}_+ - \bar{y}_-) \qquad \textbf{(32)}$$

Where $\bar{y}_+$and $\bar{y}_-$ are the average of a response corresponding to the higher and lower level of each variable.

This simple calculation leads to primary effects when the variables are evaluated separately. When the product of these variables is evaluated, we get secondary effects. If the variables are in the same level (-1 and -1 or +1 and +1), the net level is higher, equal to +1. If the variables are in the opposite level (-1 and +1 or +1 and -1), the net level is lower.

If we want to study three variables instead of two, we will have a 2^3 factorial design with 8 experiments, as shown in **Table 2**.

Table 2.*2^3 factorial design.*

ID	v_1	v_2	v_3
1	*-1*	*-1*	*-1*
2	*+1*	*-1*	*-1*
3	*-1*	*+1*	*-1*
4	*+1*	*+1*	*-1*
5	*-1*	*-1*	*+1*
6	*+1*	*-1*	*+1*
7	*-1*	*+1*	*+1*
8	*+1*	*+1*	*+1*

In this case, beyond the primary and secondary effects, we can calculate tertiary effects between the variables 123, 124, 134, and 234. For the first experiment (ID = 1), the level is lower, because -1 X -1 X -1 = -1. And for the second experiment, the level is higher because of +1 X -1 X -1 = +1.

The complete factorial design remains useful and fast until a certain point. When a large number of variables are used, the fractional factorial design is a better option. For five variables, a complete factorial design leads to $2^5 = 32$ experiments, but if a fractional factorial design is used, the number of experiments reduces by half. The following notation is used $2^{5-1} = 2^5/2 = 16$ experiments. This can be done because the number of higher-order interactions increases with the number of variables and usually these interactions have small values[88]. **Table 3** shows the 2^{5-1} factorial design. The level of variable 5 is determined by the product of the four variables.

Table 3.*2^{5-1} factorial design.*

ID	v_1	v_2	v_3	v_4	v_5
1	*-1*	*-1*	*-1*	*-1*	*+1*
2	*+1*	*-1*	*-1*	*-1*	*-1*
3	*-1*	*+1*	*-1*	*-1*	*-1*
4	*+1*	*+1*	*-1*	*-1*	*+1*
5	*-1*	*-1*	*+1*	*-1*	*-1*
6	*+1*	*-1*	*+1*	*-1*	*+1*
7	*-1*	*+1*	*+1*	*-1*	*+1*
8	*+1*	*+1*	*+1*	*-1*	*-1*
9	*-1*	*-1*	*-1*	*+1*	*-1*
10	*+1*	*-1*	*-1*	*+1*	*+1*
11	*-1*	*+1*	*-1*	*+1*	*+1*
12	*+1*	*+1*	*-1*	*+1*	*-1*
13	*-1*	*-1*	*+1*	*+1*	*+1*
14	*+1*	*-1*	*+1*	*+1*	*-1*
15	*-1*	*+1*	*+1*	*+1*	*-1*
16	*+1*	*+1*	*+1*	*+1*	*+1*

Regression models*[88]*. The regression models are used to predict the response based on the values of the most important variables. This approach allows the use of a few experiments to predict hundreds to thousands of responses, including their optimal values.

The regression quantitative models can be built by different designs: central composite design (CC), Doehlert design (D), and Box-Behnken design (BB). **Figure 14** shows these three designs for a set of three variables[89].

The central composite design is based on the factorial design and has additional axial regions. Thus, fewer experiments need to be carried out if a factorial design was previously performed.

Doehlert design consists of an asymmetric factorial design, which enables the evaluation of several variables at many levels. The main characteristic of this design is the use of a variable - the most important one - in many levels concerning the others [89].

This allows the exploration of quadratic and interaction behaviors with few experiments[88].

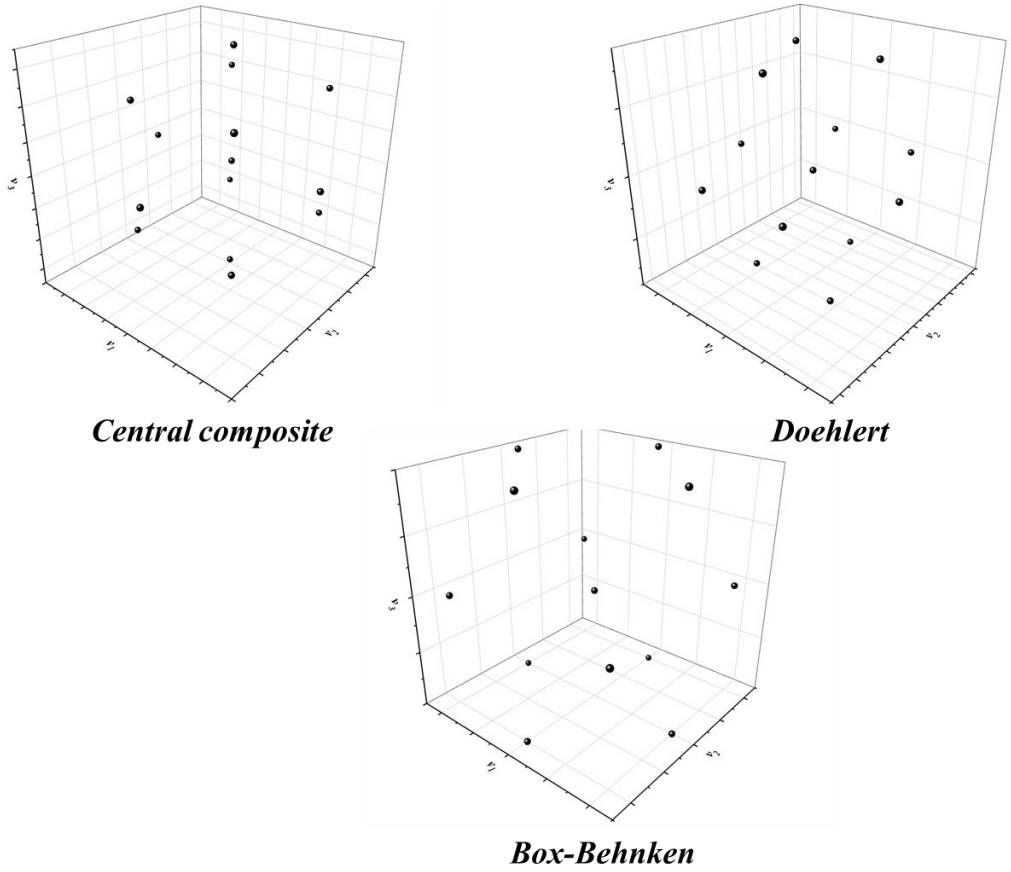


***Figure 14**. Designs for building regression models for three variables.*

The Box-Behken design consists of an incomplete factorial design and the levels are vary between -1 and +1 too[89]. The use of a design can be determined by the efficiency, the ratio between the number of coefficients by the number of experiments (see **Table 4**)[89]. The Doehlert design has the higher efficiency for designs with 2,3 and 4 variables.

***Table 4.** Characteristics of experimental designs.*

Variables	*Number of coefficients (p)*	*Number of experiments (n)*			*Efficiency (p/n)*		
		CC	*D*	*BB*	*CC*	*D*	*BB*
2	*6*	*9*	*7*	*-*	*0.67*	***0.86***	*-*
3	*10*	*15*	*13*	*13*	*0.67*	***0.77***	***0.77***
4	*15*	*25*	*21*	*25*	*0.60*	***0.71***	*0.60*

Since the design is chosen and performed, the regression model can be built by the determination of the coefficients using the least-squares fit method. This is done by the following matrixial equation[88]:

$$w = (X^T X)^{-1} X^T y \quad \textbf{(33)}$$

Where *w* is the vector containing the coefficients (b0, b1, b2…), X is the matrix containing the experimental design information and *y* is the vector containing the response.

Metrics of the regression model[87,88]**.** In this thesis, we designed simulation calculations instead of carrying out experiments at the laboratory. However, the method to evaluate the regression model is the same. The difference between experimental and simulated results is just the pure error, that we can obtain experimentally by carrying out replicates. The pure error can be used to determine the statistical significance of each coefficient because it is one part of the residual from regression. The value of pure error squared sum and residual squared sum are calculated by:

$$SS_r = SS_{pe} + SS_{lof} \quad \textbf{(34)}$$

Where SS_r is the residual sum of squares, SS_{pe} is the pure error sum of squares and SS_{lof} is the lack of fit sum of squares.

SS_r is determined by the squared difference between the predicted ($\hat{y}_i$) and experimental (y_i) responses:

$$SS_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \textbf{(35)}$$

The SS_{pe} provides information about the intrinsic error of the experiments. SS_{pe} is determined by the squared difference between the average value of the replicates ($\bar{y}_i$) by each replicate (y_{ij}):

$$SS_{pe} = \sum_{i=1}^{m} \sum_{j=1}^{n} (y_{ij} - \bar{y}_i)^2 \quad \textbf{(36)}$$

The SS_{lof} provides information about the incapacity of the model to adjust to the experimental data. SS_{pe} can be determined by the subtraction of SS_r and SS_{pe} (see again **Equation 34**).

Instead of a pure error, the simulated results have domain residuals, since we use numerical methods to solve the equations. However, these errors – residual domain and pure error – *are not equivalent.* The response obtained by simulation does not vary for each calculation, how it happens experimentally. Thus, for simulation, it is *not necessary* to carry out replicates and the value of residual sum squares from regression is exactly the value of lack of fit sum squares:

$$SS_r = SS_{lof} \quad \textbf{(37)}$$

A suitable metric to determine the statistical significance of coefficients is the rooted mean value of SS_r, calculated by:

$$RMSE = \sqrt{SS_r/n} \quad \textbf{(38)}$$

Where *RMSE* is defined as the rooted mean squared error and *n* is the number of designed experiments.

Another important metric of regression analysis is the coefficient of determination (R^2). The R^2 provides the quantity of information that regression can explain. This metric is calculated by:

$$R^2 = \frac{SS_r}{SS_t} \quad \textbf{(39)}$$

Where SS_t is the total squared sum, determined by the squared difference between the experimental values (y_i) and the mean values ($\bar{y}$):

$$SS_r = \sum_{i=1}^{n}(y_i - \bar{y})^2 \quad \textbf{(40)}$$

Validation of regression model[87,88] The data set utilized to determine the coefficients of the regression model is called *training or calibration set*. For instance, in a Doehlert design, all the experiments are from the training set.

Since the coefficients are known and there is a good correlation between simulated responses, we must validate this regression model by a second data set called *testing or validation set*. Usually, the testing set is built by the values of variables that led to the optimum response. For clarity, we will term this type of validation as *validation of optimum responses*, because we also used a difference validation termed *random validation.* These two types of validation are described below:

(i) *Validation of optimum responses:* This validation is used to confirm the values of optimum responses predicted by the regression model. Thus, the testing set consists of variables that lead to the optimum responses. The validation is carried out by setting the values of variables, which correspond to the optimum responses, in the multi-physical model and solving it. The response obtained by the simulation is, then, compared with the predicted response.

(ii) *Random validation:* This validation is carried out to evaluate the robustness of the regression model. The testing set is composed of 1/3 of experiments concerning the training set. The values of all variables are determined randomly by an operator. These values are set in the multi-physical model and the regression model and, then, compared.

The use of one or another type of validation depends on the aims of each study.

Figure 15 schematizes the process just described.

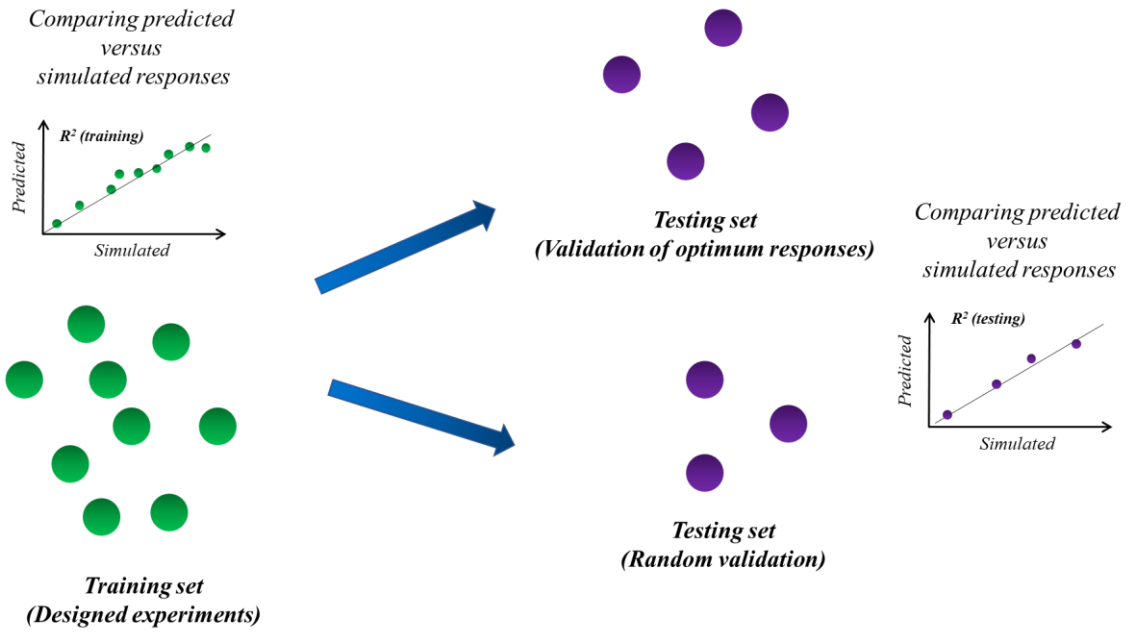


Figure 15. *Schematic representation of the process of training and testing the regression model.*

2 OBJECTIVES

This thesis is based on the hypothesis that a computational approach can provide rich and useful information for the understanding and improvement of performance of redox flow batteries. Further, this information may be used to create mitigation strategies and as guidelines for experimentalists.

Based on this hypothesis, the objective of this thesis is to propose a new method of approach to investigate problems of RFBs and suggest robust mitigation strategies for these problems. We aim to provide a useful theoretical background for the experimentalists to understand the effects of variables on RFB's performance.

3 NUMERICAL MODEL FORMULATION

This chapter brings all the details of the numerical model per component, including the governing differential equations and the boundary conditions. This chapter also provides details about the three different multi-physical models used in this work.

The numerical model described below is based on well-established studies from literature, according to the citations. However, there are some novel contributions: The modifiable domain (section 3.2), the way that self-discharge reactions are set up (section 3.10) and the development of a steady-state model to calculate voltage efficiency (section 3.11).

3.1 Domain

The domain is a 2D projection of a 3D redox flow battery composed of two electrolyte tanks, pipes, two current collectors, two porous electrodes (carbon felt), and a Nafion membrane. The current collectors, electrolyte tanks, and pipes are not present in the 2D domain. The current collector and electrolyte tanks were simulated through boundary conditions. The pipes' contribution to pumping energy consumption was added to the model using equations to describe the pressure loss. **Figure 16** shows the 3D geometry of a redox flow battery and the 2D projection in the *xy*-plane.

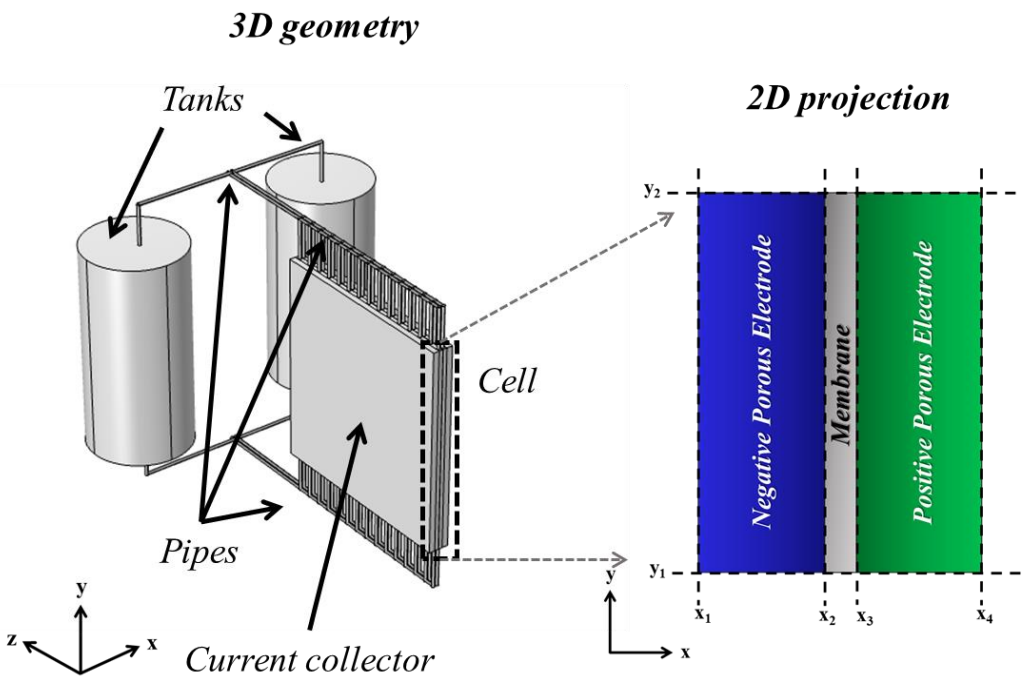


Figure 16. *3D scheme of a redox flow battery and its 2D projection in the xy-plane.*

The symmetry of the *xy*-plane along the *z*-axis allows the use of a 2D projection, without losing the physical characteristics of the device. The main benefit of using 2D models is the decreasing of the computational cost.

3.2 Modifiable domain

A modifiable domain was obtained by the Bézies Polygon setting on COMSOL Multiphysics. The purpose is to modify the cell geometry using a few parameters: the angle of modification θ, the length of residence path (H), and the inlet electrode thickness (L_{in}). This process allows the use of different cell geometry, from rectangular to trapezoid.

Each half-cell was built by four linear segments. These segments are $\underline{A_J B_j}$, $\underline{B_J C_j}$, $\underline{C_J D_j}$ and $\underline{D_J A_j}$ (where j is the negative or positive electrode) as shown in **Figure 17**.

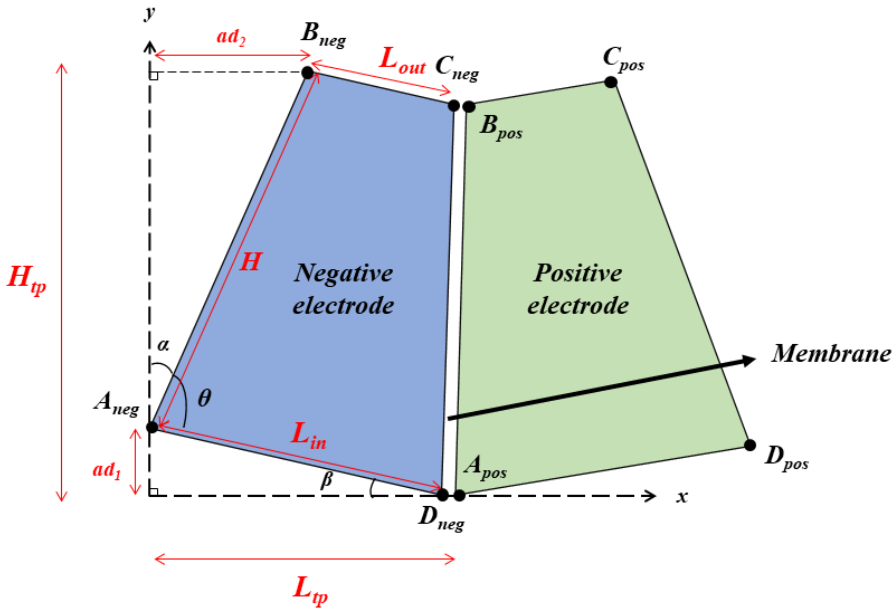


Figure 17. *Scheme of modifiable cell geometry.*

When $\theta < 90°$ the geometry represents a trapezium and when $\theta > 90°$ the geometry represents an inverted trapezium.

For the negative half-cell, the points A_{neg}, B_{neg}, C_{neg}, and D_{neg}, are:

$$\begin{cases} A_{neg} = (0, ad_1) \\ B_{neg} = (ad_2, ad_1 + H_{tp}) \\ C_{neg} = (L_{tp}, H) \\ D_{neg} = (L_{tp}, 0) \end{cases} \quad (41)$$

And, for the positive half-cell, the points A_{pos}, B_{pos}, C_{pos}, and D_{pos} are:

$$\begin{cases} A_{pos} = (L_{tp} + L_m, 0) \\ B_{pos} = (L_{tp} + L_m, H_{cell}) \\ C_{pos} = (2L_{tp} + L_m - ad_2, ad_1 + H_{tp}) \\ D_{pos} = (2L_{tp} + L_m, ad_1) \end{cases} \quad (42)$$

The geometric parameters ad_1, ad_2, L_{tp}, and H_{tp} are calculated as follows, respectively:

$$\begin{cases} ad_1 = L_{in} sin\,(\alpha) \\ ad_2 = H sin\,(\beta) \\ L_{tp} = L_{in} cos\,(\alpha) \\ H_{tp} = ad_1 + H cos\,(\beta) \end{cases} \quad \textbf{(43)}$$

Where the angles are defined as:

$$\begin{cases} \alpha = 180° - \theta \\ \beta = 90° - 2\theta \end{cases} \quad \textbf{(44)}$$

Where θ is the angle of geometric modification.

The outlet electrode thickness is calculated by:

$$L_{out} = \sqrt{\left(L_{tp} - ad_2\right)^2 + \left(H - H_{tp} - ad_1\right)^2} \quad \textbf{(45)}$$

All the above geometric parameters are determined by chosen values of θ, $L_{in,}$ and H. **Table 5** shows the default values of the geometric parameters.

Table 5. *Default values of the geometric parameters*

Symbol	*Description*	*Value (unit)*[ref]
H	*Cell height (length of residence path)*	10 cm [90]
w_{cell}	*Cell width*	10 cm [90]
L_{cc}	*Current collector thickness*	6 mm [90]
L_{in}	*Inlet electrode thickness*	4 mm [90]
L_m	*Membrane thickness*	180 μm [90]
θ	*Angle of modification*	90°

3.3 Premises

The following premises were applied:

1. The cell is isothermal;

2. Fluids are considered incompressible;

3. The properties of charge and mass transfer are assumed to be isotropic and do not change in different charge states;

4. Mass transport is described using a diluted solution model;

5. In electrolytic tanks, the solutions are instantly homogenized.

3.4 Porous electrodes

As the electrode is porous there is a solid volumetric fraction (*ε-1*) and a liquid volumetric fraction equal to the porosity of the electrode (*ε*). For the vanadium redox flow batteries, the electrolyte is composed of aqueous solutions of V^{2+}/V^{3+} (as sulfate salts) and H_2SO_4 in the negative half-cell and VO^{2+}/VO_2^{+} (as sulfate salts) and H_2SO_4 in the positive half-cell. The first dissociation of sulfuric acid is complete:

$$H_2SO_4 \rightarrow H^+ + HSO_4^- \quad \textbf{(46)}$$

The second dissociation is incomplete:

$$HSO_4^- \rightleftharpoons H^+ + SO_4^{2-} \quad \textbf{(47)}$$

Thus, it is necessary to use a term that represents this last dissociation[74]:

$$S_d = k_d \left(\frac{c^e_{H^+} - c^e_{HSO_4^-}}{(c^e_{H^+} - c^e_{HSO_4^-})} - \beta \right) \quad \textbf{(48)}$$

Where S_d is the source of species (mol m^{-3} s^{-1}), k_d is the coefficient of dissociation, c^e_k ($k \in \{H^+, HSO_4^-\}$) is the concentration of species in the electrolyte and β is the dissociation degree determined experimentally[91].

The mass conservation for each species is expressed as follows:

$$\frac{\partial}{\partial t}(\varepsilon c_i^e) + \nabla \cdot \vec{N}_i^e = -S_i \quad \textbf{(49)}$$

Where ε is the porosity of the electrode, c_i^e ($i \in \{V^{2+}, V^{3+}, VO^{2+}, VO_2^+, H^+, HSO_4^-\}$) is the bulk concentration for each species, $\vec{N}_i^e$ is the flux and S_i is the source term.

This equation is not applied to SO_4^{2-}, because its concentration is calculated by neutrality condition:

$$\sum_i z_i c_i^e = 0 \quad \textbf{(50)}$$

Where z_i is the valence of each species.

For the flux, $\vec{N}_i^e$, the Nernst-Planck equation is applied:

$$\vec{N}_i^e = -D_i^{eff} \nabla c_i^e - z_i u_i^e c_i^e F \nabla \phi_l^e + \vec{v} c_i^e \quad \textbf{(51)}$$

Where ϕ_l^e is the ionic potential (liquid phase), D_i^{eff} is the effective diffusion coefficient, u_i^e is the ionic mobility and $\vec{v}$ is the velocity.

The effective diffusion coefficient is estimated by the Bruggemann correction:

$$D_i^{eff} = \varepsilon^{\frac{3}{2}} D_i \quad \textbf{(52)}$$

The velocity is determined by Darcy's law:

$$\vec{v} = -\frac{\kappa}{\mu^{e,j}} \nabla p \quad \textbf{(53)}$$

Where μ^e is the dynamic viscosity of the electrolyte, p is the pressure and κ is the permeability of electrode, which is determined by the Kozeny-Carman equation:

$$\kappa = \frac{d_f^2}{180} \frac{\varepsilon^3}{(1-\varepsilon)^2} \quad \textbf{(54)}$$

Where d_f is the carbon electrode fiber diameter.

The charge conservation is expressed as follows:

$$\nabla \cdot \vec{\imath}_l + \nabla \cdot \vec{\imath}_s = 0 \quad \textbf{(55)}$$

Where i_l is the ionic current density (electrolyte) and $\vec{\imath}_s$ is the electronic current density (electrode). This equation couples the transport of ionic species to the electrochemical reaction and the electronic current[74].

The ionic current density is described by Faraday's law:

$$\vec{\imath}_e = F \sum_i z_i \vec{N}_i^e \quad \textbf{(56)}$$

The electronic current density is described by Ohm's law:

$$\vec{\imath}_s = \sigma_s^{eff} \nabla \phi_s^e \quad \textbf{(57)}$$

Where σ_s^{eff} is the effective conductivity of the porous electrode, calculated by the Bruggemann correction:

$$\sigma_s^{eff} = (1-\varepsilon)^{3/2} \sigma_s \quad \textbf{(58)}$$

Where $\boldsymbol{\sigma_s}$ is the electronic conductivity of the electrode.

Table 6 shows the default values of the porous electrode parameters.

Table 6. *Default values of the porous electrode*

Symbol	*Description*	*Value (unit)*[ref]
a	*Specific surface area*	$2 \times 10^6\ m^{-1}$ [90]
β	*HSO_4^- degree of dissociation*	0.25 [74]
d_f	*Mean pore diameter*	$100.6\ \mu m$ [74]
$D_{H^+}^e$	*H^+ diffusion coefficient*	$9.3 \times 10^{-9}\ m^2\ s^{-1}$ [92]
$D_{HSO_4^-}^e$	*HSO_4^- diffusion coefficient*	$1.3 \times 10^{-9}\ m^2\ s^{-1}$ [92]
$D_{SO_4^{2-}}^e$	*SO_4^{2-} diffusion coefficient*	$1.1 \times 10^{-9}\ m^2\ s^{-1}$ [92]
$D_{V^{2+}}^e$	*V(II) diffusion coefficient*	$2.4 \times 10^{-10}\ m^2\ s^{-1}$ [93]
$D_{V^{3+}}^e$	*V(III) diffusion coefficient*	$2.4 \times 10^{-10}\ m^2\ s^{-1}$ [93]
$D_{VO^{2+}}^e$	*V(IV) diffusion coefficient*	$3.9 \times 10^{-10}\ m^2\ s^{-1}$ [93]
$D_{VO_2^+}^e$	*(V) diffusion coefficient*	$3.9 \times 10^{-10}\ m^2\ s^{-1}$ [93]
ε	*Electrode porosity*	0.93 [74]
k_d	*HSO_4^- dissociation reaction rate constant*	$1 \times 10^4\ s^{-1}$ [74]
σ_{cc}	*Electronic conductivity of current collector*	$1000\ S\ m^{-1}$ [90]
σ_s^{eff}	*Effective conductivity of the electrode*	$66.7\ S\ m^{-1}$ [74]

3.5 Electrochemical reactions

The Nernst potential for the electrochemical reactions are:

$$E^n = E^{0,n} + \frac{RT}{F} ln\left(\frac{c_{V^{3+}}}{c_{V^{2+}}}\right) \quad \textbf{(59)}$$

$$E^p = E^{0,p} + \frac{RT}{F} ln\left(\frac{c_{VO_2^+} c_{H^+_{pos}}^2}{c_{VO^{2+}}}\right) \quad \textbf{(60)}$$

The concentration of species is used instead of activity due to the premise of diluted solution.

The local currents, $i_{loc}^{n,}$ and i_{loc}^{p}, are governed by the Butler-Volmer equation:

$$i_{loc}^n = aFk^n (c_{V^{2+}}^e)^{(1-\alpha^n)} (c_{V^{3+}}^e)^{\alpha^n} \left[\frac{c_{V^{2+}}^s}{c_{V^{2+}}^e} exp\left(\frac{(1-\alpha^n)F\eta^n}{RT}\right) - \frac{c_{V^{3+}}^s}{c_{V^{3+}}^e} exp\left(\frac{-\alpha^n F\eta^n}{RT}\right)\right] \quad \textbf{(61)}$$

$$i_{loc}^p = aFk^p (c_{VO^{2+}}^e)^{(1-\alpha^p)} (c_{VO_2^+}^e)^{\alpha^p} \left[\frac{c_{VO^{2+}}^s}{c_{VO^{2+}}^e} exp\left(\frac{(1-\alpha^p)F\eta^p}{RT}\right) - \frac{c_{VO_2^+}^s}{c_{VO_2^+}^e} exp\left(\frac{-\alpha^p F\eta^p}{RT}\right)\right] \quad \textbf{(62)}$$

Where k is the rate constant, a is the specific surface area, α is the charge transfer coefficient, s denotes the surface concentration, e denotes the bulk concentration and η is the overpotential, defined as:

$$\eta^j = \phi_l^e - \phi_s^e - E^{0,j} \quad \textbf{(63)}$$

Where $E^{0,j}$ is the standard reduction potential. The obtained overpotential is separated into concentration overpotential and activation overpotential [94]:

$$n^j = n_c^j + n_a^j \quad \textbf{(64)}$$

The activation overpotential for $\alpha = 0.5$ is calculated by [95]:

$$\eta_a^n = -\frac{2RT}{F} sinh^{-1}\left(\frac{i_{loc}^n}{2Fk^n\sqrt{c_{V^{2+}}^e c_{V^{3+}}^e}}\right) \quad \textbf{(65)}$$

$$\eta_a^p = \frac{2RT}{F} sinh^{-1}\left(\frac{i_{ioc}^p}{2Fk^{pos}\sqrt{c_{VO^{2+}}^e c_{VO_2^+}^e}}\right) \quad \textbf{(66)}$$

Since n^j is known, the concentration overpotential for each cell is just the subtraction of activation overpotential from the total overpotential.

The concentration overpotential arises from the local transfer resistance from the bulk to the active reaction surface. This concentration overpotential is added to the Butler-Volmer equations by the quotient between surface and bulk concentrations. The surface concentrations are calculated by the species balance at active reaction surface[76]. The ionic current at the active reaction surface for each species is[76]:

$$i_{active\ surface}^j = FD_i\left(\frac{c_i^e - c_i^s}{d_f/2}\right) \quad \textbf{(67)}$$

Which is equal to each local current described by Butler-Volmer equations **54** and **55**. This mass balance leads to the following equations:

For V^{2+}:

$$FD_{V^{2+}}\left(\frac{c_{V^{2+}}^e - c_{V^{2+}}^s}{d_f/2}\right) = aFk^n (c_{V^{2+}}^e)^{(1-\alpha^n)} (c_{V^{3+}}^e)^{\alpha^n}\left[\frac{c_{V^{2+}}^s}{c_{V^{2+}}^e} exp\left(\frac{(1-\alpha^n)F\eta^n}{RT}\right) - \frac{c_{V^{3+}}^s}{c_{V^{3+}}^e} exp\left(\frac{-\alpha^n F\eta^n}{RT}\right)\right] \quad \textbf{(68)}$$

For V^{3+}:

$$FD_{V^{3+}}\left(\frac{c_{V^{3+}}^e - c_{V^{3+}}^s}{d_f/2}\right) = aFk^n (c_{V^{2+}}^e)^{(1-\alpha^n)} (c_{V^{3+}}^e)^{\alpha^n}\left[\frac{c_{V^{2+}}^s}{c_{V^{2+}}^e} exp\left(\frac{(1-\alpha^n)F\eta^n}{RT}\right) - \frac{c_{V^{3+}}^s}{c_{V^{3+}}^e} exp\left(\frac{-\alpha^n F\eta^n}{RT}\right)\right] \quad \textbf{(69)}$$

For VO^{2+}:

$$FD_{VO^{2+}}\left(\frac{c^e_{VO^{2+}}-c^s_{VO^{2+}}}{d_f/2}\right)=$$
$$aFk^p(c^e_{VO^{2+}})^{(1-\alpha^p)}(c^e_{VO_2^+})^{\alpha^p}\left[\frac{c^s_{VO^{2+}}}{c^e_{VO^{2+}}}exp\left(\frac{(1-\alpha^p)F\eta^p}{RT}\right)-\frac{c^s_{VO_2^+}}{c^e_{VO_2^+}}exp\left(\frac{-\alpha^pF\eta^p}{RT}\right)\right] \quad \textbf{(70)}$$

For VO_2^+:

$$FD_{VO_2^+}\left(\frac{c^e_{VO_2^+}-c^s_{VO_2^+}}{d_f/2}\right)=$$
$$-aFk^p(c^e_{VO^{2+}})^{(1-\alpha^p)}(c^e_{VO_2^+})^{\alpha^p}\left[\frac{c^s_{VO^{2+}}}{c^e_{VO^{2+}}}exp\left(\frac{(1-\alpha^p)F\eta^p}{RT}\right)-\frac{c^s_{VO_2^+}}{c^e_{VO_2^+}}exp\left(\frac{-\alpha^pF\eta^p}{RT}\right)\right] \quad \textbf{(71)}$$

Isolating the surface concentration for each species, we get:

For V^{2+}:

$$c^s_{V^{2+}}=\frac{c^e_{V^{2+}}+A_1(c^e_{V^{2+}}+c^e_{V^{3+}})}{1+A_1+B_1} \quad \textbf{(72)}$$

For V^{3+}:

$$c^s_{V^{3+}}=\frac{c^e_{V^{3+}}+B_1(c^e_{V^{2+}}+c^e_{V^{3+}})}{1+A_1+B_1} \quad \textbf{(73)}$$

Where:

$$A_1=exp\left(\frac{-\alpha^nF\eta^n}{RT}\right)\frac{k^nd_f}{2D_{V^{2+}}}\left(\frac{c^e_{V^{2+}}}{c^s_{V^{2+}}}\right)^{\alpha^n} \quad \textbf{(74)}$$

$$B_1=exp\left(\frac{(1-\alpha^n)F\eta^n}{RT}\right)\frac{k^nd_f}{2D_{V^{3+}}}\left(\frac{c^e_{V^{3+}}}{c^s_{V^{3+}}}\right)^{(1-\alpha^n)} \quad \textbf{(75)}$$

For VO^{2+}:

$$c^s_{VO^{2+}}=\frac{c^e_{VO^{2+}}+A_2(c^e_{VO^{2+}}+c^e_{VO_2^+})}{1+A_2+B_2} \quad \textbf{(76)}$$

For VO_2^+:

$$c_{VO_2^+}^s = \frac{c_{VO_2^+}^e + B_2(c_{VO^{2+}}^e + c_{VO_2^+}^e)}{1 + A_2 + B_2} \quad (77)$$

Where:

$$A_2 = exp\left(\frac{-\alpha^p F \eta^p}{RT}\right) \frac{k^p d_f}{2D_{VO^{2+}}} \left(\frac{c_{VO^{2+}}^e}{c_{VO^{2+}}^s}\right)^{\alpha^p} \quad (78)$$

$$B_1 = exp\left(\frac{(1-\alpha^p) F \eta^p}{RT}\right) \frac{k^p d_f}{2D_{VO_2^+}} \left(\frac{c_{VO_2^+}^e}{c_{VO_2^+}^s}\right)^{(1-\alpha^p)} \quad (79)$$

Table 7 shows the electrochemical model parameters.

***Table 7**. Electrochemical model parameters*

Symbol	*Description*	*Value (unit)[ref]*
$E^{0,n}$	*Standard reduction potential (negative)*	$-0.255\ V$ [90]
$E^{0,p}$	*Standard reduction potential (positive)*	$1.004\ V$ [90]
k^n	*Reaction rate constant (negative electrode)*	$7 \times 10^{-8}\ m\ s^{-1}$ [74]
k^p	*Reaction rate constant (positive electrode)*	$2.5 \times 10^{-8}\ m\ s^{-1}$ [74]
α^n	*Charge transfer coefficient (negative)*	0.5
α^p	*Charge transfer coefficient (positive)*	0.5

The charge transfer coefficients are set as 0.5 as an approximation to an easy calculation of activation overpotentials (**Equations 65 and 66**).

3.6 Membrane

Since the Nafion membrane is not ideally selective to H^+[43], the transport of vanadium species must be considered. The Nernst-Planck equation is applied and the concentration of HSO_4^{2-} is calculated by the electroneutrality condition, considering the fixed charges at membrane:

$$z_{fix} c_{fix} + \sum_i z_i c_i^m = 0 \quad (80)$$

Where z_{fix} and c_{fix} are the valence and concentration of sulfonyl groups fixed at the membrane.

The convection at the membrane is calculated by an alternative form of Schogl's equation[74]:

$$\vec{v}_m = -\frac{\kappa_p}{\mu_m}\nabla p - \frac{\kappa_\phi}{\mu_m} c_{fix} F \left(\nabla \phi_l^m + \frac{F \sum z_i D_i^m \nabla c_i^m}{\frac{F^2}{RT} \sum z_i^2 D_i^m c_i^m} \right) \quad \textbf{(81)}$$

Where κ_p is the hydraulic permeability, μ_m is the viscosity of water, κ_ϕ is the electrokinetic permeability, $\nabla \phi_l^m$ is the potential gradient across membrane and D_i^m ($i \in \{V^{2+}, V^{3+}, VO^{2+}, VO_2^+, H^+, HSO_4^-\}$ denotes the diffusion coefficient of each species in the membrane.

The first term of **Equation 81** represents the osmosis across the membrane caused by pressure gradients between the two electrolytes. The second term represents the electroosmosis caused by the drag of solvent in the membrane.

The flux of vanadium species in the electrode interface region is calculated by [74]:

$$N_i^{er} = \frac{-D_i^{eff}(c_i^e - c_i^{junc})}{\delta} - z_i u_i^e \frac{\left(c_i^e + c_i^{junc}\right)}{2} K_\phi \frac{(\phi_l^e - \phi_l^m)}{\delta} \quad \textbf{(82)}$$

Where c_i^{junc} represents the concentration of species at membrane|electrolyte junction, δ is the thickness of interfacial region, u_i^e is the ionic mobility in the electrolyte and K_ϕ is a fitting parameter that represents the percentage of the total potential-jump occurring in the electrolyte interface region.

The flux of vanadium species in the membrane interface region is calculated by [74]:

$$N_i^{mr} = \frac{-D_i^m(c_i^{junc} - c_i^m)}{\delta} - z_i u_i^m \frac{\left(c_i^{junc} + c_i^m\right)}{2} (1 - K_\phi) \frac{(\phi_l^e - \phi_l^m)}{\delta} \quad \textbf{(83)}$$

The junction concentration, c^{junc}, is calculated by setting the fluxes at the interface equal and solving for c^{junc}.

Table 8 shows the default values of the membrane.

Table 8. *Default values of the membrane*

Symbol	*Description*	*Value (unit)[ref]*
$D_{H^+}^m$	*H^+ diffusion coefficient*	$3.35 \times 10^{-9}\ m^2\ s^{-1}$ [96]
$D_{HSO_4^-}^m$	*HSO_4^- diffusion coefficient*	$4{,}0 \times 10^{-11}\ m^2\ s^{-1}$ [97]
$D_{SO_4^{2-}}^m$	*SO_4^{2-} diffusion coefficient*	$2.2 \times 10^{-10}\ m^2\ s^{-1}$ [98]
$D_{V^{2+}}^m$	*V(II) diffusion coefficient*	$8{,}8 \times 10^{-12}\ m^2\ s^{-1}$ [48]
$D_{V^{3+}}^m$	*V(III) diffusion coefficient*	$3{,}2 \times 10^{-12}\ m^2\ s^{-1}$ [48]
$D_{VO^{2+}}^m$	*V(IV) diffusion coefficient*	$6{,}8 \times 10^{-12}\ m^2\ s^{-1}$ [48]
$D_{VO_2^+}^m$	*(V) diffusion coefficient*	$5{,}9 \times 10^{-12}\ m^2\ s^{-1}$ [48]
$c_{H^+,0}^m$	*H^+ initial concentration*	$4772{,}5\ mol\ m^{-3}$ [74]
c_{fix}	*Fixed acid concentration*	$2000\ mol\ m^{-3}$ [99]
z_{fix}	*Fixed acid charge*	-1 [74]
ε^m	*Porosity of the membrane*	$0{,}135$ [99]
κ_p	*Hydraulic permeability*	$1{,}58 \times 10^{-18}\ m^2$ [100]
κ_ϕ	*Electrokinetic permeability*	$1{,}13 \times 10^{-20}\ m^2$ [74]
μ_m	*Viscosity of water*	$8.90 \times 10^{-4}\ Pa\ s$

3.7 Electrolyte tanks

The variation of quantity of species, in mol, in the electrolyte tanks follow the ODE:

$$\frac{dn_i^t}{dt} = \varepsilon w_{cell}\left(\int_0^{L_{out}} c_i v dx - \int_0^{L_{in}} c_i v dx\right), \quad n_i^t(0) = c_{i,0} V_0^t \qquad \textbf{(84)}$$

Where n_i^t is the quantity, in mol, of species *i* in the electrolyte tank, *v* is the linear velocity inside the cell, $c_{i,0}$ is the initial concentration of species *i*, and V^t_0 is the initial volume of each tank.

Since there is a electrolyte transfer between the half-cells, caused by the convection across membrane, a variation of volume is observed in the electrolyte tanks.The variation of volume in each electrolyte tank follows the ODE:

$$\frac{dV^{t,j}}{dt} = j\varepsilon w_{cell} H v_{m,x}, \quad V^{t,j}(0) = V_0^t \qquad \textbf{(85)}$$

Where $V^{t,j}$ is the volume of each tank for each half-cell and j is the unit value of semi-cell representation (-1 for the negative half-cell and +1 for the positive half-cell).

Thus, the concentration in the tanks follow the equation:

$$c_i^t = \frac{n_i^t}{V^{t,j}} \quad \textbf{(86)}$$

3.8 Pumping energy consumption

The pumping energy consumption was considered by calculating the power required for electrolyte circulation[55,56,101]:

$$P_{pump} = \frac{\Delta p_{total}\omega}{\alpha_{pump}} \quad \textbf{(87)}$$

Where α_{pump} is the pump efficiency and Δp_{total} is the total pressure loss, understood as the sum of the pressure loss caused by the flow in the porous electrode (carbon felt) and the pressure loss caused by the friction of the electrolyte in the pipes:

$$\Delta p_{total} = \Delta p_{electrode}^{neg} + \Delta p_{electrode}^{pos} + \Delta p_{pipes}^{neg} + \Delta p_{pipes}^{pos} \quad \textbf{(88)}$$

The pressure loss related to the friction in the pipes was calculated by the Darcy-Weisbach equation[56]:

$$\Delta p_{pipes}^{j} = n_{pipes} f_d^j \frac{L_{pipe}\rho^j {v^j}^2}{2d_{pipe}} \quad \textbf{(89)}$$

Where n_{pipes} is the number of pipes for each half-cell, L_{pipe} is the pipe length, ρ^j is the fluid density for each half-cell, v^j is the inlet flow velocity for each half-cell, d_{pipe} is the pipe diameter and f_d^j is the Darcy friction factor for each half-cell, given by:

$$f_d^j = \frac{16\pi\mu^j d_{pipe}}{\rho^j \omega} \quad \textbf{(90)}$$

Where μ^j is the dynamic viscosity of each electrolyte and ω is the volumetric flow.

The number of pipes was assumed based on the cell width considering the number of channels spaced four times the diameter of the pipes:

$$n_{pipes} = \frac{w_{cell}}{4d_{pipe}} \quad \textbf{(91)}$$

The pressure loss through the porous electrode is derived from Darcy's law and can be determined by the pressure difference between the flow inlet and outlet:

$$\Delta p^{j}_{electrode} = \left(\frac{\int p^{in} dx}{L_{in}} - \frac{\int p^{out} dx}{L_{out}}\right) \quad \textbf{(92)}$$

Where p^{in} and p^{out} are the inlet and outlet pressure, respectively.

Table 9 shows the default values of the hydraulic model parameters.

Table 9. *Hydraulic model parameters*

Symbol	*Description*	*Value (unit)[ref]*
L_{pipe}	*Pipe length*	$10H_{cell}$
d_{pipe}	*Pipe diameter*	$L_e - 0.5\ mm$
p^{out}	*Outlet pressure*	$1\ atm$
α_{pump}	*Pump efficiency*	0.9 [101]
μ^n	*Dynamic viscosity (negative)*	$2.5\ mPa \cdot s$ [74]
μ^p	*Dynamic viscosity (positive)*	$5.0\ mPa \cdot s$ [74]
ρ^n	*Density of negative electrolyte*	$1300\ kg\ m^{-3}$ [74]
ρ^p	*Density of positive electrolyte*	$1350\ kg\ m^{-3}$ [74]
ω	*Volumetric flow*	$20\ mL\ min^{-1}$

3.9 Initial and boundary conditions

The boundary conditions are assumed in the boundaries x_1, x_2, x_3, x_4, y_1, and y_2 in the domain (see **Figure 18**).

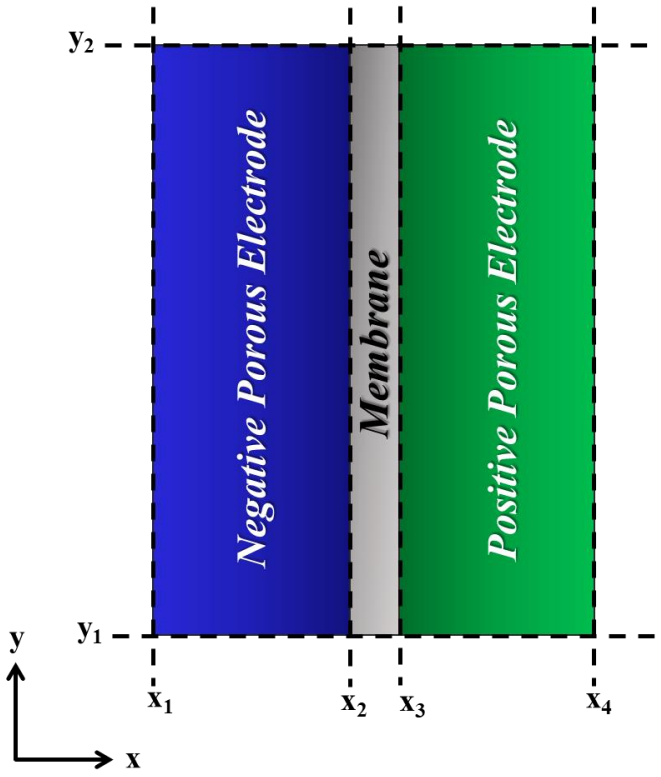


Figure 18. *2D domain representation.*

The conservation of flux and ionic current density are boundary conditions that reflect the passage of species across the electrolyte/membrane and membrane/electrolyte interface, that is:

$$\begin{cases} \vec{n} \cdot \vec{i}_i^{\,e} = \vec{n} \cdot \vec{i}_i^{\,m} \\ \vec{n} \cdot \vec{N}_i^{\,e} = \vec{n} \cdot \vec{N}_i^{\,m} \end{cases} \quad x = x_2, x_3 \qquad \textbf{(93)}$$

Where $\vec{n}$ is the normal vector.

There is no species flux in the current collectors and the top and bottom of the membrane, that is:

$$0 = \begin{cases} \vec{n} \cdot \vec{N}_i & y = y_1, y_2 \\ \vec{n} \cdot \vec{N}_i & x = x_1, x_4 \end{cases} \qquad \textbf{(94)}$$

And, for the flow outlet ($y = y_2$):

$$\begin{cases} p = p^{out} \\ -\vec{n} \cdot D_i^{eff} \nabla c_i^e = 0 \end{cases} \quad \textbf{(95)}$$

The potential at the current collector boundary ($x = x_1$) was zero:

$$\phi_s^{cc} = 0 \quad \textbf{(96)}$$

The inlet and outlet and the membrane/electrolyte were taken to be electrically insulated:

$$-\vec{n} \cdot \vec{i}_s = -\vec{n} \cdot \vec{i}_l \qquad x = x_2, x_3; y = y_1, y_2 \quad \textbf{(97)}$$

The concentration in the inlet is equal to the concentration of species in the electrolyte tanks:

$$c_i^t = c_i^{in} \qquad y = y_1 \quad \textbf{(98)}$$

The flux of species across membrane due cross-contamination is given by:

$$-\vec{n} \cdot \vec{N}_i^m = -\vec{n} \cdot \vec{N}_i^{mr} \qquad x = x_2, x_3 \quad \textbf{(99)}$$

The flux of species across electrolytes due to cross-contamination is given by:

$$-\vec{n} \cdot \vec{N}_i^e = \vec{n} \cdot \vec{N}_i^{er} \qquad x = x_2, x_3 \quad \textbf{(100)}$$

The charge/discharge cycle in galvanostatic mode may be obtained by SoC cut-off or voltage cut-off.

For SoC cut-off, the following boundary condition was applied:

$$-\vec{n} \cdot \vec{i}_s^{cc} = \begin{cases} i_{appl} & SoC - SoC_{max} < 0 \\ -i_{appl} & SoC - SoC_{min} > 0 \end{cases} \quad x = x_4 \quad \textbf{(101)}$$

Where i_{appl} is the current density applied, SoC_{max} is the maximum state of charge (to the charge step) and SoC_{min} is the minimum state of charge (to the discharge step).

For voltage cut-off:

$$-\vec{n}\cdot\vec{i}_s^{cc} = \begin{cases} i_{appl} & V - V_{max} < 0 \\ -i_{appl} & V - V_{min} > 0 \end{cases} \quad x = x_4 \qquad \textbf{(102)}$$

Where V_{max} is the maximum cell voltage (to the charge step) and V_{min} is the minimum cell voltage (to the discharge step).

Table 10 shows the default values for the initial conditions.

Table 10. *Initial conditions*

Symbol	*Description*	*Value (unit)[ref]*
SoC_{max}	*Maximum state of charge*	*0.85*
SoC_{min}	*Minimum state of charge*	*0.15*
V^T	*Electrolyte volume in each half-cell*	50 mL
$c^n_{H^+,0}$	*H^+ initial concentration (negative)*	4447.5 $mol\ m^{-3}$ [74]
$c^p_{H^+,0}$	*H^+ initial concentration (positive)*	5097.5 $mol\ m^{-3}$ [74]
$c^n_{HSO_4^-,0}$	*HSO_4^- initial concentration (negative)*	2668.5 $mol\ m^{-3}$ [74]
$c^p_{HSO_4^-,0}$	*HSO_4^- initial concentration (positive)*	3058.5 $mol\ m^{-3}$ [74]
$c_{V^{2+},0}$	*V(II) initial concentration*	150 $mol\ m^{-3}$
$c_{V^{3+},0}$	*V(III) initial concentration*	850 $mol\ m^{-3}$
$c_{VO^{2+},0}$	*V(IV) initial concentration*	850 $mol\ m^{-3}$
$c_{VO_2^+,0}$	*V(V) initial concentration*	150 $mol\ m^{-3}$
p_0	*Initial pressure in electrodes*	0 atm
T	*Temperature*	298.15 K

3.10 Self-discharge reactions

To simulate the self-discharge reactions, the species V^{2+} and V^{3+} are also present in the positive electrode, and the species VO^{2+} and VO_2^+ are present in the negative electrode as V^{2+}_{pos}, V^{3+}_{pos}, VO^{2+}_{neg} and $VO_2{}^{+}_{neg}$ respectively. The self-discharge reactions were set in y_2 axis for simplification.

For the negative electrode, the following equations are applied:

$$(i) - \vec{n}\cdot\vec{N}_i^e = -2N^e_{H^+} - N^e_{V^{2+}} + 3N^e_{V^{3+}} - N^e_{VO^{2+}} \qquad \textbf{(103)}$$

In this case, the $VO^{2+}{}_{neg}$ limits the reaction; thus, for n $c^e_{VO^{2+}{}_{neg}} = 0 \Longrightarrow -\vec{n} \cdot \vec{N}_i^e = 0$.

$$(ii) - \vec{n} \cdot \vec{N}_i^e = -4N^e_{H^+} - 2N^e_{V^{2+}} + 3N^e_{V^{3+}} - N^e_{VO_2^+} \quad \textbf{(104)}$$

For $c^e_{VO_2^+{}_{neg}} = 0 \implies -\vec{n} \cdot \vec{N}_i^e = 0$.

For the positive electrode, the following equations are applied:

$$\text{(iii)} -\vec{n} \cdot \vec{N}_i^e = -2N^e_{H^+} - N^e_{V^{2+}} + 3N^e_{VO^{2+}} - 2N^e_{VO_2^+} \quad \textbf{(105)}$$

For $c^e_{V^{2+}{}_{pos}} = 0 \Longrightarrow -\vec{n} \cdot \vec{N}_i^e = 0$.

$$(iv) - \vec{n} \cdot \vec{N}_i^e = -N^e_{V^{3+}} + 2N^e_{VO^{2+}} - 2N^e_{VO_2^+} \quad \textbf{(106)}$$

For $c^e_{V^{3+}{}_{pos}} = 0 \Longrightarrow -\vec{n} \cdot \vec{N}_i^e = 0$.

3.11 The multi-physical models

The level of theory of a numerical model depends on the kind of response we want to extract from it. For example, for the determination of cell potential, it is not necessary to simulate the cross-contamination between half-cells. First, the cell potential of a single cycle in a numerical model without cross-contamination is almost equal to a numerical model with cross-contamination. That is, the cross-contamination has effects only for several charge/discharge cycles. And second, the more complete a model, the higher the computational cost. Then, is not reasonable to use a complete model just to obtain the cell potential for a single cell. Because of this, we adapted the numerical model described in this chapter according to the purpose of each study. These models are described below.

(1) VRFB capacity loss model (time-dependent). This multi-physical model aims to calculate the capacity loss of a vanadium redox flow battery across several galvanostatic charge/discharge cycles All the equations related to this model were described in this chapter. This model is the most complete used in this work.

The validation of this multi-physical model was performed by comparing the experimental and simulated discharge capacity. The experimental values were taken from literature [102].

Figure 19 shows the comparison between simulated and experimental discharge capacity.

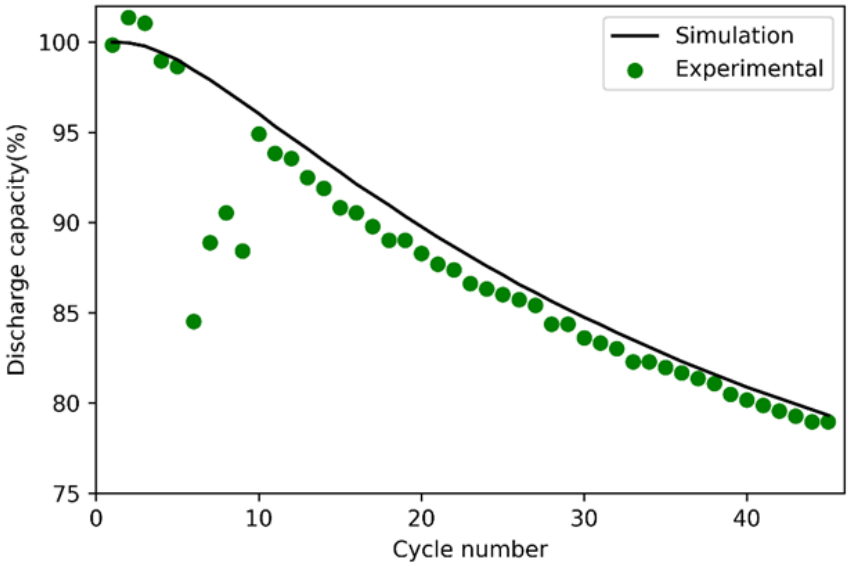


Figure 19. *Comparison between simulated and experimental discharge capacity. Experimental data from* [102].

The simulated response captures the key trends of the experimental results. The correction parameter used was the electrokinetic permeability, k_{ϕ}, just like in the multi-physical model developed by Knehr et. al.[74]. This correction parameter is modify so that experimental results agree with the simulate results, considering a possible erros of approximation in the model.

(2) VRFB cell potential model (time-dependent). This multi-physical model aims to calculate the cell voltage of a vanadium redox flow battery in a single galvanostatic

charge/discharge cycle. All equations related to this model were described in this section, but membrane domain.

A simpler approach was employed from the membrane since the main purpose of this model was the study of the effects on performance caused by overpotentials and pumping energy consumption.

The ionic current was given by:

$$\vec{\imath}_m = -\sigma_l^m \nabla \phi_m \qquad \textbf{(107)}$$

Where σ_l^m is the ionic conductivity of the membrane.

And there were no sources of ionic current in the membrane:

$$\nabla \cdot \vec{\imath}_m = 0 \qquad \textbf{(108)}$$

The Donnan potential condition was applied:

$$E^m = \phi_l^e - \phi_l^m = \frac{RT}{F} ln\left(\frac{[H^+]_e}{[H^+]_m}\right) \qquad \textbf{(109)}$$

Where $[H^+]_e$ is the proton concentration in electrolyte and $[H^+]_m$ is the proton concentration in the membrane.

The validation of this multi-physical model was performed by comparing the experimental and simulated cell voltage. The experimental values were taken from literature [74].

Figure 20 shows the comparison between simulated and experimental cell voltage.

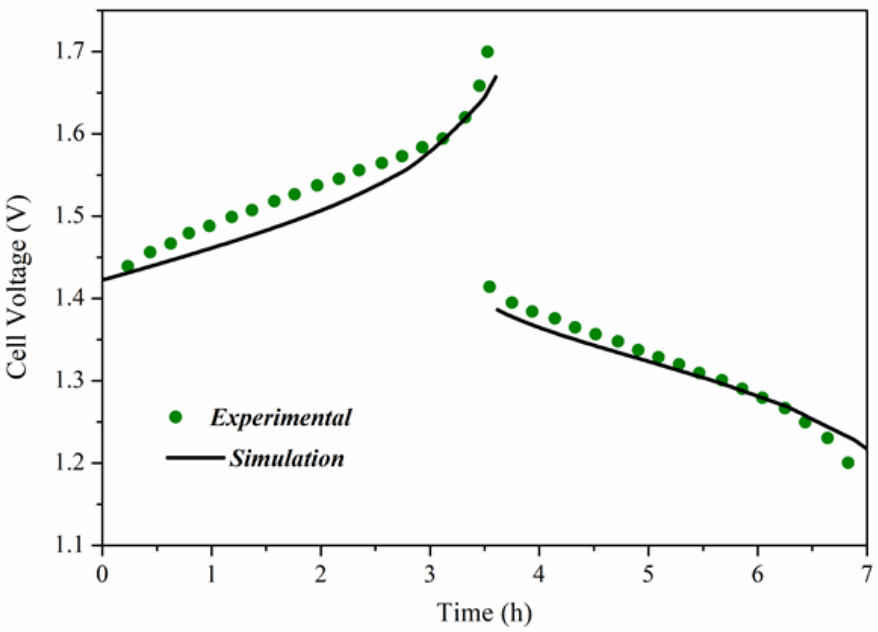


Figure 20. *Comparison between simulated and experimental cell voltage. Experimental data from* [74].

Despite the difference in concavity in charging procedure, the simulated response captures the key trends of the experimental results. No correction parameter was used.

(3) RFB general model (stationary). This multi-physical model is an adaptation of the time-dependent VRFB cell voltage model with two differences: (i) it is a general model to portray any kind of redox flow battery and (ii) it is stationary, aiming at for the reduction of computational costs.

It was considered a single-electron transfer reaction for both half-cells:

$$R^{x}_{anolyte} \rightleftharpoons O^{x-1}_{anolyte} + e^{-} \quad \textbf{(110)}$$

$$O^{y}_{catholyte} + e^{-} \rightleftharpoons R^{y+1}_{catholyte} \quad \textbf{(111)}$$

The Nernst equations are, respectively:

$$E^{n} = E^{0,n} + \frac{RT}{F} ln\left(\frac{c_{O^{x-1}_{anolyte}}}{c_{R^{x}_{anolyte}}}\right) \quad \textbf{(112)}$$

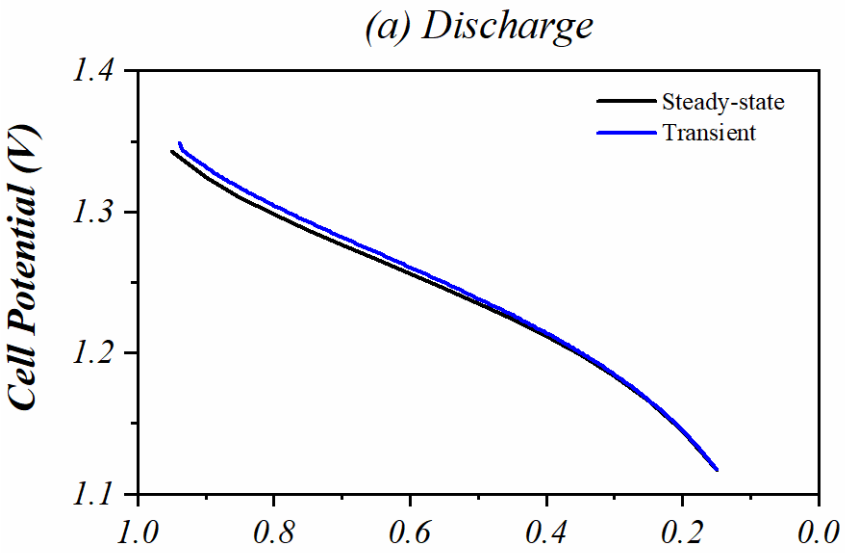
$$E^{p} = E^{0,p} + \frac{RT}{F} ln\left(\frac{c_{O^{y}_{catholyte}}}{c_{R^{y+1}_{catholyte}}}\right) \quad \textbf{(113)}$$

The steady-state condition is achieved by setting the concentration of active species in the electrolyte tanks as constant. Thus, the cell is operating in a single SoC. It is possible to simulate different SoCs just by varying the concentrations of active species in the tanks. That is,

$$\begin{cases} c_{O_{anolyte}} = c_t(1 - SoC) \\ c_{R_{anolyte}} = c_t SoC \end{cases} \quad y = y_1 \tag{114}$$

$$\begin{cases} c_{O_{catholyte}} = c_t SoC \\ c_{R_{catholyte}} = c_t(1 - SoC) \end{cases} \quad y = y_1 \tag{115}$$

The validation of this model was carried out by comparison of cell potential versus the state of charge (SoC) from the VRFB transient model with the VRFB steady-state model (see **Figure 21**).



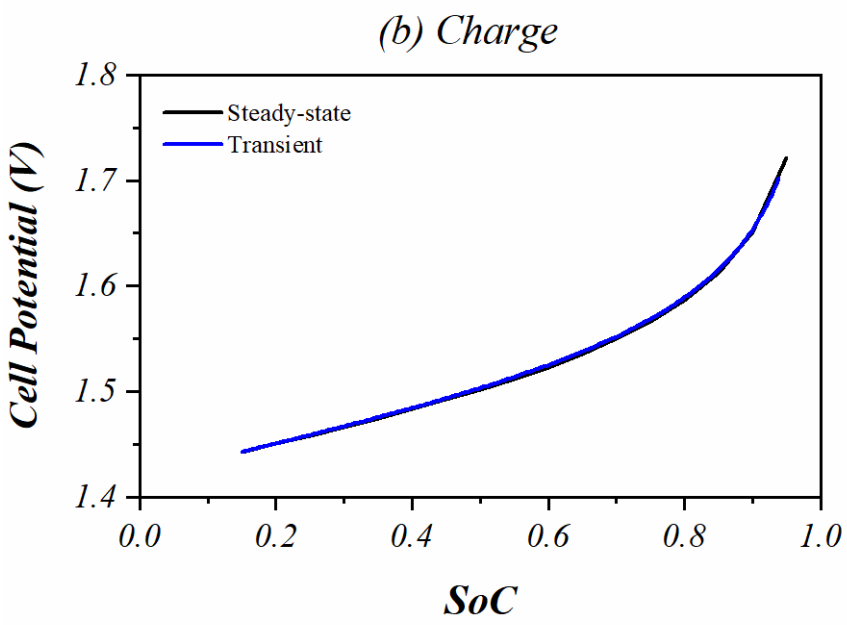


Figure 21. *Comparison between steady-state and transient multi-physical model by cell potential versus the state of charge (SoC) for (a) charge and (b) discharge.*

The charging potential is identical for the steady-state and transient VRFB model. However, the discharging potential is slightly different for high SoCs. This may be associated with the contribution of proton concentration on the positive Nernst potential:

$$E^p = E^{0,p} + \frac{RT}{F} ln\left(\frac{c_{VO_2^+} c_{H^+_{pos}}^2}{c_{VO^{2+}}}\right) \quad \textbf{(116)}$$

In the transient model, the proton concentration in each half-cell changes over time in the inlet channels due to crossover across membrane and chemical equilibria (see equation below). In the steady-state model, the proton concentration does not change over time in the inlet channels leading to the observed difference in cell potential.

$$HSO_4^- \rightleftharpoons H^+ + SO_4^{2-} \quad \textbf{(117)}$$

4 METHODOLOGY

This chapter elucidates the computational method of approach step-by-step, describes the numerical methods used to solve the multi-physical models, and detail the three carried out computational studies: ***Study 1:*** *Understanding how operating conditions affect optimal geometry for vanadium redox flow batteries;* ***Study 2:*** *A robust mitigation strategy for capacity loss in VRFBs under different operating conditions;* ***Study 3:*** *Insights into the effects of active species properties on the performance of redox flow batteries.*

4.1 Method of Approach

The general method of approach consists of three levels: experimental results, multi-physical model, and chemometrics analysis (see **Figure 22**).

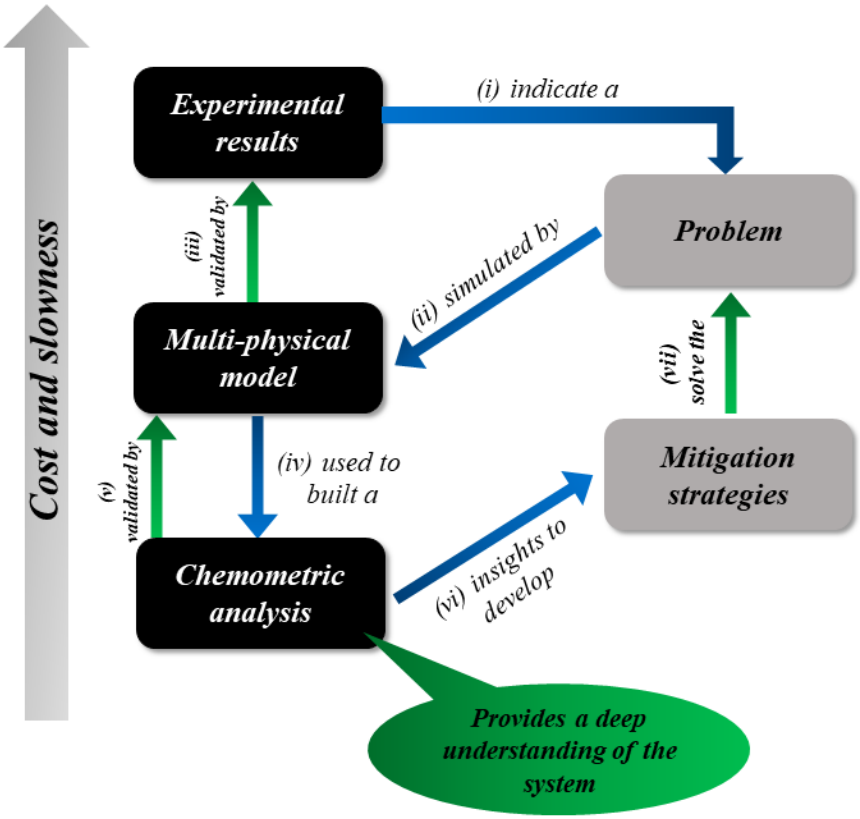


Figure 22. *Scheme of the general method of approach.*

The steps are described below:

(i) Experimental indication of a problem. Previous studies indicate a drawback related to the system under study. (e.g. capacity loss caused by cross-contamination).

(ii) Development of a multi-physical model. The problem indicated by the previous step may be described by a suitable multi-physical model, aiming at the save of time and cost to carry out several computational experiments.

(iii) Validation of the multi-physical model. The numerical model must be able to portray reality with confidence. This step is done by the comparison of simulated results with experimental data from literature.

(iv) Use of chemometric analysis. The use of chemometric analysis, as factorial design and regression analysis, allows a deep understanding of the system, the screening of important variables, and a fast prevision of hundreds to thousands of responses in a short time.

(v) Validation of the chemometric analysis. In the case of regression analysis, it is necessary to verify the capacity of the regression model to predict the responses of the computational experiments. This is done by comparing the predicted responses with the simulated data.

(vi and vii) Insights to develop a mitigation strategy. The knowledge obtained in the previous step is used to develop a mitigation strategy for the problem of interest.

An additional step is validating this mitigation strategy in an experimental setup. However, this work focus on developing this computational approach.

4.2 Numerical methods

The equations were implemented and solved using the finite element method with COMSOL Multiphysics® package using the interfaces: Tertiary Current Distribution, Second Current Distribution, Darcy's Law, Events, and Global ODEs and DAESs. The calculations were run on a PC with a 3.5 GHz processor i7 and 131 GB of RAM. The approximated time for the computational experiments is 12 h, 70 min, and 2 min for the

VRFB capacity loss model, VRFB cell potential model, and RFB general model, respectively. The relative tolerance was set to 1×10^{-3}.

4.3 Studies

4.3.1 Study 1: Understanding how operating conditions affect optimal geometry for vanadium redox flow batteries

Since it is known that geometric conditions affect the voltage efficiency of VRFBs, a systematic study can provide useful information for us to understand these effects. Beyond studying geometric parameters, we want to evaluate how these parameters are related to the operating conditions since these affect the performance too.

Based on the proposed method of approach, we identified the problem, developed and validated the multi-physical model and, used the multivariate regression analysis to predict thousands of responses based on the conditions under study. **Table 11** shows the previous characteristics (step (i) to (iv)) of Study 1 based on this approach

Table 11. *Previous characteristics of Study 1 based on the method of approach.*

Step	*Description*	*Checklist*
(i)	*Experimental indication of a problem*	*Energy efficiency loss caused by overpotential and pumping energy consumption*
(ii)	*Development of a multi-physical model.*	*VRFB cell potential model (SoC cut-off)*
(iii)	*Validation of the multi-physical model.*	*Figure 20 (successfully validated)*
(iv)	*Use of chemometric analysis.*	*Multivariate regression analysis*

The problem of this study is the energy efficiency loss caused by overpotential (ohmic and concentration) and by pumping energy consumption. The multi-physical model used is the VRFB cell potential model (time-dependent), which was successfully validated comparing simulated and experimental results (see **Figure 20**). And the multivariate regression analysis was the chemometric analysis used. The remaining steps ((v) to (vii)) were discussed in the RESULTS AND DISCUSSION section.

Design of experiments. The experimental space was built from five variables: volumetric flow (ω), applied current density (i_{appl}), inlet electrode thickness (L_{in}) without

compression, length of residence path (H), and the ratio between outlet electrode thickness and inlet electrode thickness (L_{out}/L_{in}) without compression. See **Figure 23** to identify the geometric parameters. Aiming for a robust exploration of the experimental space, the Doehlert design was used to select the combination of the variables (see **Table 12**).

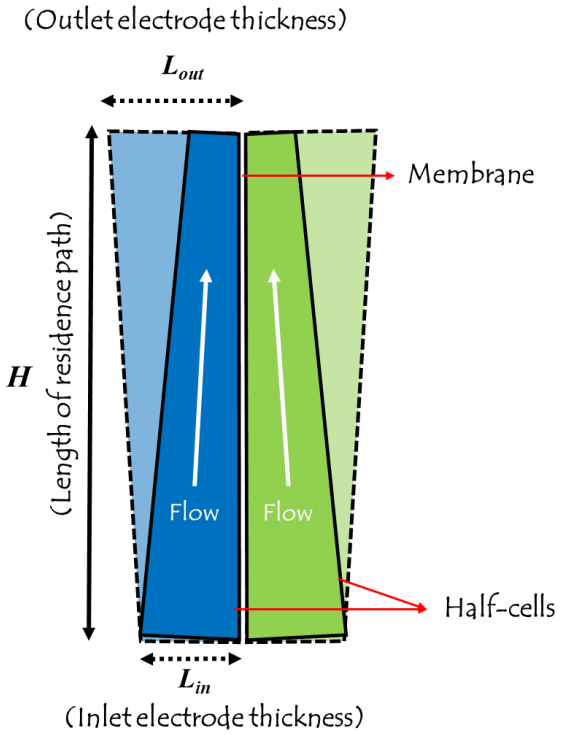


Figure 23. *Schematic representation of geometric parameters.*

Table 12*. Doehlert design for five variables of Study 1 – codified*

ID	L_e	i_{appl}	ω	H	L_{out}/L_{in}
1	*0*	*0*	*0*	*0*	*0*
2	*1*	*0*	*0*	*0*	*0*
3	*0.5*	*0.866*	*0*	*0*	*0*
4	*0.5*	*0.289*	*0.817*	*0*	*0*
5	*0.5*	*0.289*	*0.204*	*0*	*0.791*
6	*0.5*	*0.289*	*0.204*	*0.775*	*0*
7	*-1.00*	*0*	*0*	*0*	*0*
8	*-0.50*	*-0.866*	*0*	*0*	*0*
9	*-0.50*	*-0.289*	*-0.817*	*0*	*0*
10	*-0.50*	*-0.289*	*-0.204*	*0*	*-0.791*
11	*-0.50*	*-0.866*	*-0.204*	*-0.775*	*-0.158*
12	*0.5*	*-0.289*	*0*	*0*	*0*
13	*0.5*	*-0.289*	*-0.817*	*0*	*0*
14	*0.5*	*-0.289*	*-0.204*	*0*	*-0.791*

ID	L_e	i_{appl}	ω	*H*	L_{out}/L_{in}
15	*0.5*	*-0.289*	*-0.204*	*-0.775*	*-0.158*
16	*-0.5*	*0.866*	*0*	*0*	*0*
17	*0*	*0.577*	*-0.817*	*0*	*0*
18	*0*	*0.577*	*-0.204*	*0*	*-0.791*
19	*0*	*0.577*	*-0.204*	*-0.775*	*0*
20	*-0.5*	*0.289*	*0.817*	*0*	*0*
21	*0*	*-0.577*	*0.817*	*0*	*0*
22	*0*	*0*	*0.613*	*0*	*-0.791*
23	*0*	*0*	*0.613*	*-0.775*	*-0.158*
24	*-0.5*	*0.289*	*0.204*	*0*	*0.791*
25	*0*	*-0.577*	*0.204*	*0*	*0.791*
26	*0*	*0*	*-0.613*	*0*	*0.791*
27	*0*	*0*	*0*	*-0.775*	*0.633*
28	*-0.5*	*0.289*	*0.204*	*0.775*	*0.158*
29	*0*	*-0.577*	*0.204*	*0.775*	*0.158*
30	*0*	*0*	*-0.613*	*0.775*	*0.158*
31	*0*	*0*	*0*	*0.775*	*-0.633*

The values used in the simulation was determined by:

$$a = 0.5(a_+ - a_-)x + a_{CP} \qquad \textbf{(118)}$$

Where a is the decodified value of a variable, x is the codified value, a_+ is the upper value ($x = +1$), a_- is the lower value ($x = -1$) and a_{CP} is the central point value ($x = 0$).

For the studied variables, these values are:

Table 13. *Codification parameters*

parameter	L_e *(mm)*	i_{appl} *(mA cm^{-2})*	ω *(L min^{-1})*	*H (cm)*	L_{out}/L_{in}
a_+	*8*	*200*	*1*	*50*	*1.6*
a_-	*3*	*20*	*0.15*	*10*	*0.4*
a_{CP}	*5*	*100*	*0.5*	*30*	*1*

Table 14 shows the decodified variables for the Doehlert design.

Table 14. *Doehlert design of Study 1 - decodified*

ID	L_{in} *(mm)*	i_{appl} *(mA cm^{-2})*	ω *(L min^{-1})*	*H (cm)*	L_{out}/L_{in}
1	*5.00*	*100.0000*	*0.5000*	*30.0*	*1.0000*
2	*7.50*	*100.0000*	*0.5000*	*30.0*	*1.0000*
3	*6.25*	*177.9400*	*0.5000*	*30.0*	*1.0000*
4	*6.25*	*126.0100*	*0.8472*	*30.0*	*1.0000*
5	*6.25*	*126.0100*	*0.5867*	*30.0*	*1.4746*

ID	*L_{in} (mm)*	*i_{appl} (mA cm^{-2})*	*ω (L min^{-1})*	*H (cm)*	*L_{out}/L_{in}*
6	*6.25*	*126.0100*	*0.5867*	*45.5*	*1.0948*
7	*2.50*	*100.0000*	*0.5000*	*30.0*	*1.0000*
8	*3.75*	*22.0600*	*0.5000*	*30.0*	*1.0000*
9	*3.75*	*73.9900*	*0.1528*	*30.0*	*1.0000*
10	*3.75*	*73.9900*	*0.4133*	*30.0*	*0.5254*
11	*3.75*	*22.0600*	*0.4133*	*14.5*	*0.9052*
12	*6.25*	*73.9900*	*0.5000*	*30.0*	*1.0000*
13	*6.25*	*73.9900*	*0.1528*	*30.0*	*1.0000*
14	*6.25*	*73.9900*	*0.4133*	*30.0*	*0.5254*
15	*6.25*	*73.9900*	*0.4133*	*14.5*	*0.9052*
16	*3.75*	*177.9400*	*0.5000*	*30.0*	*1.0000*
17	*5.00*	*151.9300*	*0.1528*	*30.0*	*1.0000*
18	*5.00*	*151.9300*	*0.4133*	*30.0*	*0.5254*
19	*5.00*	*151.9300*	*0.4133*	*14.5*	*0.9052*
20	*3.75*	*126.0100*	*0.8472*	*30.0*	*1.0000*
21	*5.00*	*48.0700*	*0.8472*	*30.0*	*1.0000*
22	*5.00*	*100.0000*	*0.7605*	*30.0*	*0.5254*
23	*5.00*	*100.0000*	*0.7605*	*14.5*	*0.9052*
24	*3.75*	*126.0100*	*0.5867*	*30.0*	*1.4746*
25	*5.00*	*48.0700*	*0.5867*	*30.0*	*1.4746*
26	*5.00*	*100.0000*	*0.2395*	*30.0*	*1.4746*
27	*5.00*	*100.0000*	*0.5000*	*14.5*	*1.3798*
28	*3.75*	*126.0100*	*0.5867*	*45.5*	*1.0948*
29	*5.00*	*48.0700*	*0.5867*	*45.5*	*1.0948*
30	*5.00*	*100.0000*	*0.2395*	*45.5*	*1.0948*
31	*5.00*	*100.0000*	*0.5000*	*45.5*	*0.6202*

Regression Analysis. The designed conditions were used for calculation in the multi-physical model 2. The voltage efficiency, battery efficiency, pumping energy consumption, concentration overpotential, and ohmic overpotential were calculated. For the overpotentials, the response corresponds to the mean value in the discharge procedure.

The responses were individually submitted to a quadratic regression aiming at the determination of the following coefficients (in bold):

$$Response = \boldsymbol{w_0} + \boldsymbol{w_{01}} L_{in} + \boldsymbol{w_{02}} i_{appl} + \boldsymbol{w_{03}} \omega + \boldsymbol{w_{04}} H + \boldsymbol{w_{05}} (L_{out}/L_{in}) + \boldsymbol{w_{11}} L_{in}^2 + \boldsymbol{w_{12}} L_{in} i_{appl} + \boldsymbol{w_{13}} L_{in} \omega + \boldsymbol{w_{14}} L_{in} H + \boldsymbol{w_{15}} L_{in} (L_{out}/L_{in}) + \boldsymbol{w_{22}} {i_{appl}}^2 + \boldsymbol{w_{23}} i_{appl} \omega + \boldsymbol{w_{24}} i_{appl} H + \boldsymbol{w_{25}} i_{appl} (L_{out}/L_{in}) + \boldsymbol{w_{33}} \omega^2 + \boldsymbol{w_{34}} \omega H + \boldsymbol{w_{35}} \omega (L_{out}/L_{in}) + \boldsymbol{w_{44}} H^2 + \boldsymbol{w_{45}} H (L_{out}/L_{in}) + \boldsymbol{w_{55}} (L_{out}/L_{in})^2 \quad \textbf{(119)}$$

Where w_0 is the intercept, w_{0i} are linear coefficients, w_{ii} are quadratic coefficients and w_{ij}, with i ≠ j, are interaction coefficients.

The statistical significance of the coefficients was calculated based on rooted mean squared error for regression with a significance level of 0.05.

Performance metrics. The cell voltage was calculated as follows[103]:

$$V_{cell} = E^p - E^n + (\phi_l^p - \phi_l^n) + \eta_a^p + \eta_c^p - \eta_a^n - \eta_c^n + \eta_o \quad \textbf{(120)}$$

Where *E* is the Nernst potential and the difference $\phi_l^p - \phi_l^n$ is the potential loss across the membrane. The subscripts *a*, *c* and *o* describe activation, concentration, and ohmic, respectively.

The ohmic overpotential, η_o, is given by the contributions of the current collector, porous electrode, and membrane:

$$\eta_o = 2(\eta_o)_s^{cc} + (\eta_o)_l^{e,n} + (\eta_o)_l^{e,p} + (\eta_o)^m \quad \textbf{(121)}$$

Where the superscripts *cc*, *e*, and *m* describe the current collector, electrode, and membrane, respectively. The subscripts *s* and *l* describe the solid and liquid states, respectively.

These components were expressed as:

$$\begin{cases} (\eta_o)_s^{cc} = i_{appl} \frac{L_{cc}}{\sigma_s^{cc}} \\ (\eta_o)_l^{e,j} = i_{appl} \frac{L_{avg}}{\varepsilon^{3/2} \sigma_l^{e,j}} \\ (\eta_o)^m = i_{appl} \frac{L_m}{\sigma_l^m} \end{cases} \quad \textbf{(122)}$$

Where L_{avg} is the average electrode thickness calculated as:

$$L_{avg} = \frac{A_g}{H} \quad \textbf{(123)}$$

Where A_g is the area of the geometric 2D model for each half-cell.

The coulombic efficiency (CE) was calculated based on the times of charging and discharging procedures. For this study, the charge/discharge cycles are symmetrical and there are no parasitic reactions, then, the calculated CE for all experiments is 100%.

$$CE = \frac{t_d}{t_c} = 1 \quad \textbf{(124)}$$

Where t_d is the discharge time and t_c is the charge time.

The voltage efficiency (VE) was calculated as follows:

$$VE = \frac{\bar{V}_d}{\bar{V}_c} = \frac{\left(\int_0^{t_d} V_d dt\right)/t_d}{\left(\int_0^{t_c} V_c dt\right)/t_c} \quad \textbf{(125)}$$

Where V_d is the charging voltage and V_c is the discharge voltage.

The energy efficiency (EE) was calculated as:

$$EE = CE \cdot VE \quad \textbf{(126)}$$

Since CE is equal to 1, the EE is:

$$EE = VE \quad \textbf{(127)}$$

The pumping energy consumption is defined as:

$$E_{pump} = \int_0^t P_{pump}\, dt \quad \textbf{(128)}$$

Where P_{pump} is the pump power.

The battery efficiency (BE), which includes the pumping energy consumption, was calculated as:

$$BE = \frac{Q_d \bar{V}_d - E_{pump}}{Q_c \bar{V}_c - E_{pump}} \quad \textbf{(129)}$$

Where Q_d is the retained charge in the discharging procedure and Q_c is the charge applied in the charging procedure.

4.4.2 Study 2: A robust mitigation strategy for capacity loss in VRFBs under different operating conditions

Since capacity loss is a big issue for the operational life of VRFBs, we designed a study to investigate the variables that affect capacity loss and we studied the optimum conditions of volume transfer between tanks aiming at its mitigation.

Based on the proposed method of approach, we identified the problem, developed and validated the multi-physical model and, used a factorial design to screen the variables and regression analysis to predict thousands of responses based on the conditions under stud. **Table 15** shows the previous characteristics (steps (i) to (iv)) of Study 2 based on the method of approach.

Table 15. *Previous characteristics of Study 2 based on the method of approach.*

Step	*Description*	*Checklist*
(i)	*Experimental indication of a problem*	*Capacity loss caused by cross-contamination*
(ii)	*Development of a multi-physical model.*	*VRFB capacity loss model (voltage cut-off)*
(iii)	*Validation of the multi-physical model.*	*Figure 19 (successfully validated)*
(iv)	*Use of chemometric analysis.*	*Factorial design and regression analysis*

The problem of this study is the capacity loss caused by the cross-contamination between the half-cells of a VRFB. The multi-physical model used is the VRFB capacity loss model (time-dependent), which was successfully validated comparing simulated and experimental results (see **Figure 19**). And the factorial design and the regression analysis were the used chemometric tools. The remaining steps ((v) to (vii)) were discussed in the RESULTS AND DISCUSSION section.

Factorial design. A 2^3 complete factorial design was performed analysing the following variables: applied current density (i_{appl}), active species concentration (c_{act}) and volumetric flow (ω). **Table 16** describes these computational experiments.

Table 16. *2^3 factorial design of Study 2.*

ID	*i_{appl} (mA cm^{-2})*	*c_{act} (mol L^{-1})*	*ω (L min^{-1})*
1	*50*	*1.10*	*15*
2	*100*	*1.10*	*15*
3	*50*	*1.80*	*15*
4	*100*	*1.80*	*15*
5	*50*	*1.10*	*30*
6	*100*	*1.10*	*30*
7	*50*	*1.8*	*30*
8	*100*	*1.8*	*30*

The discharging capacity was calculated by[74]:

$$C_d = \frac{t_{d,n^{th}}}{t_{d,1^{st}}} \times 100\% \quad \textbf{(130)}$$

Where $t_{dis,n^{th}}$ is the discharge time of the n^{th} cycle and $t_{dis,1^{st}}$ is the discharge time of the first cycle.

The capacity loss was calculated by:

$$CL = |100\% - C_d^{th}| \quad \textbf{(131)}$$

Where $C_d{}^{th}$ is the discharge capacity at nth cycle.

The potential cut-off for each experiment was determined by the SoC cut-off of 0.9 and 0.1 for the first cycle for charging and discharging procedures, respectively. The concentration of supporting electrolyte was 3 mol L^{-1}. And the capacity loss rate was determined by:

$$CL_{rate} = \sum_1^i \frac{dC_{d,i}}{dt} \times \frac{1}{n_{cycles}} \quad \textbf{(132)}$$

Where n_{cycles} is the total number of cycles.

Determination of electrolyte viscosity (μ^j). The concentration of active species influences the electrolyte viscosity in each half-cell. And the viscosity influences the capacity loss rate due to the contribution to the convective flow across the membrane. To determine the suitable viscosity for both negative and positive electrolytes, a regression analysis was performed with data available in literature[72,73]. The studied variables were the SoC, concentration of supporting electrolyte, and concentration of active species. The

parameters values of viscosity set up in each experiment of **Table 16** were the mean values of viscosity in each SoC.

The responses (positive and negative viscosity) were individually submitted to a quadratic regression aiming at the determination of the following coefficients (in bold):

$$\mu^j = \boldsymbol{w_0} + \boldsymbol{w_{01}}SoC + \boldsymbol{w_{02}}c_{act} + \boldsymbol{w_{03}}c_{sup} + \boldsymbol{w_{04}}T + \boldsymbol{w_{11}}(SoC)^2 + \boldsymbol{w_{12}}SoC\,c_{act} + \boldsymbol{w_{13}}SoC\,c_{sup} + \boldsymbol{w_{14}}SoCT + \boldsymbol{w_{22}}c_{act}^2 + \boldsymbol{w_{23}}c_{act}c_{sup} + \boldsymbol{w_{24}}c_{act}T + \boldsymbol{w_{33}}c_{sup}^2 + \boldsymbol{w_{34}}c_{sup} + \boldsymbol{w_{44}}T^2 \quad \textbf{(133)}$$

Where c_{sup} is the concentration of active species and T is the temperature of the electrolyte.

This regression model was validated by cross-validation. The data set was split into five subsets.

Model adaptation to include the volumetric transfer between tanks. To simulate the volumetric transfer between tanks it is necessary to modify the ODEs related to the tanks. In this case, the number of species, in mol, in each tank also depends on the flux of species between the tanks. For the VO_2^+, VO^{2+}, VO^{2+}_{neg}, and $VO_2^+{}_{neg}$ the equation turns to:

$$\frac{dn_i^t}{dt} = \varepsilon w_{cell}\left(\int_0^{L_{out}} c_i v dx - \int_0^{L_{in}} c_i v dx\right) - j\omega_{br}c_i^t \quad \textbf{(134)}$$

Where ω_{br} is the volumetric flow between the tanks.

For H^+ and HSO_4^- in both half-cells, the equation turns to:

$$\frac{dn_i^t}{dt} = \varepsilon w_{cell}\left(\int_0^{L_{out}} c_i v dx - \int_0^{L_{in}} c_i v dx\right) - ij\omega_{br}c_i^t \quad \textbf{(135)}$$

Where i is the factor of current: +1 for charge and -1 for discharge.

For contaminant species in the positive electrode, that is, V^{2+}_{pos} and V^{3+}_{pos}, the ODE remains unchanged.

The volume of the tank turns to:

$$\frac{dV^{t,j}}{dt} = \varepsilon w_{cell} H v_{m,x} - j\omega_{bt} \quad \textbf{(136)}$$

Regression analysis. Since the variables that affect capacity loss are known, a regression analysis may be performed with these variables and the velocity between electrolytes tanks to evaluate how the discharge capacity behaves in these conditions. The experimental space was built from three variables in a Doehlert design: velocity between reservoirs (v_{br}), applied current density (i_{appl}) and active species concentration (c_{act}).

The capacity loss was submitted to a quadratic regression aiming at the determination of the following coefficients (in bold):

$$CL = \boldsymbol{w_0} + \boldsymbol{w_{01}} v_p + +\boldsymbol{w_{02}} i_{appl} + \boldsymbol{w_{03}} c_{act} + \boldsymbol{w_{11}} v_p^2 + \boldsymbol{w_{12}} v_p i_{appl} + \boldsymbol{w_{13}} v_p c_{act} + \boldsymbol{w_{22}} i_{appl}^2 + \boldsymbol{w_{23}} i_{appl} c_{act} + \boldsymbol{w_{33}} c_{act}^2 \quad \textbf{(137)}$$

Table 17 shows the design of experiments for this regression analysis.

Table 17. *Doehlert design for three variables of Study 2.*

ID	v_p *(XVc)*	i_{appl} *(mA cm^{-2})*	c_{act} *(mol L^{-1})*
1	*5.50*	*80.00*	*1.450*
2	*10.00*	*80.00*	*1.450*
3	*7.75*	*114.64*	*1.450*
4	*1.00*	*80.00*	*1.450*
5	*3.25*	*45.36*	*1.450*
6	*7.75*	*45.36*	*1.450*
7	*3.25*	*114.64*	*1.450*
8	*7.75*	*91.56*	*1.777*
9	*3.25*	*68.44*	*1.123*
10	*7.75*	*68.44*	*1.123*
11	*5.50*	*103.08*	*1.123*
12	*3.25*	*91.56*	*1.777*
13	*5.50*	*56.92*	*1.777*
14	*5.50*	*80.00*	*1.695*
15	*5.50*	*80.00*	*1.205*

The real value of v_{br} is calculated based on a volumetric coefficient (V_c). That is, the value of v_{br} depends on the volumetric flow (ω) and the diameter of pipe between the two reservoirs:

$$v_{br} = v_p V_c = v_p \times 10^{-5} \frac{\omega}{d_{br}^2} \quad \textbf{(138)}$$

4.4.3 Study 3: Insights into the effects of active species properties on the voltage efficiency of redox flow batteries

The choice of active species is one of the most important steps to achieve RFBs with high energy density, high power density, and high performance. It is already understood how the properties of active species affect energy density and power density. However, there is a lack of understanding of how the properties of active species affect the performance of RFBs. To obtain insights from this issue, we designed a study to investigate which properties affect the RFBs' performance and which ones are the most important.

Based on the proposed method of approach, we identified the problem, developed and validated the multi-physical model and, used a factorial design to screen the variables. **Table 18** shows the complete characteristics (steps (i) to (iv)) of Study 3 based on the method of approach.

Table 18. *Complete characteristics of Study 3 based on the method of approach.*

Step	*Description*	*Checklist*
(i)	*Experimental indication of a problem*	*Energy efficiency loss caused by choice of active species*
(ii)	*Development of a multi-physical model.*	*RFB general model (SoC cut-off)*
(iii)	*Validation of the multi-physical model.*	*Figure 21 (successfully validated)*
(iv)	*Use of chemometric analysis.*	*Factorial design*

The problem of this study is the energy efficiency loss caused by the choice of different active species. The multi-physical model used is the RFB general model (stationary), which was successfully validated comparing results from steady-state and transient models (see **Figure 21**). The factorial design was used as the chemometric tool. Thus, for this study, there was no need for the validation of regression analysis (Step (v) from the method of approach). And there is no mitigation strategy to be developed (Steps (vi) and (vii) from the method of approach) since the aim of the study is the obtaining of information about the studied conditions.

Analyzing real systems. The first step of this study was exploring the already used anolytes and catholytes in the general steady-state multi-physical model. Five active

species were selected for the study (see **Table 19**). The studied variables were the diffusion coefficient (*D*), the standard electrochemical rate constant (k^0), and the standard reduction potential (E^0).

Table 19. *Active species selected for Study 3*

ID^{ref}	*Class*	$D\ (X10^{-6}\ cm^2\ s^{-1})^a$	$k^0\ (X10^{-6}\ cm\ s^{-1})^b$	$E^0\ (V\ vs\ NHE)^c$
Methyl viologen[104]	*Anolyte*	*25.7*	*280*	*-0.45*
4-OH-TEMPO[104]	*Catholyte*	*29.5*	*260*	*0.80*
FcNCl[26]	*Catholyte*	*3.7*	*3.6*	*0.61*
V(II)/V(III)[74]	*Anolyte*	*2.4*	*7.0*	*-0.25*
V(IV)/V(V) [74]	*Catholyte*	*3.9*	*2.5*	*1.00*

[a] Diffusion coefficient in water at 25°C, [b] Glassy carbon electrode, [c] Approximated from half-wave potential.

Three combinations were studied (see **Table 20**). Each system was simulated in ten different applied current densities, from 50 to 140 mA cm^{-2}.

Table 20. *Combination of active species for the Study 3*

Anolyte	*Catholyte*	*Cell potential (V)*
Methyl viologen	*4-OH-TEMPO*	*1.25*
Methyl viologen	*FcNCl*	*1.06*
V(II)/V(III)	*V(IV)/V(V)*	*1.25*

Factorial design. A 2^{5-1} fractional factorial design was performed with the variables: standard electrochemical rate constant (k^0), diffusion coefficient (*D*), the standard cell potential (*E*), applied current density (i_{appl}), the concentration of active species (c_{act}) and volumetric flow (ω) (see **Table 21**).

Table 21. *2^{5-1} factorial design for six variables of Study 3.*

ID	$log(k^0)^a$	$D\ (X10^{-5}\ cm^2\ s^{-1})^a$	$E(V)^b$	$i_{appl}\ (mA\ cm^{-2})$	$c_{act}\ (mol\ L^{-1})$	$\omega\ (mL\ min^{-1})$
1	*-5.3*	*0.25*	*0.8*	*50*	*1*	*5*
2	*-2.3*	*0.25*	*0.8*	*50*	*1*	*20*
3	*-5.3*	*2.5*	*0.8*	*50*	*1*	*20*
4	*-2.3*	*2.5*	*0.8*	*50*	*1*	*5*
5	*-5.3*	*0.25*	*1.2*	*50*	*1*	*20*
6	*-2.3*	*0.25*	*1.2*	*50*	*1*	*5*
7	*-5.3*	*2.5*	*1.2*	*50*	*1*	*5*
8	*-2.3*	*2.5*	*1.2*	*50*	*1*	*20*
9	*-5.3*	*0.25*	*0.8*	*80*	*1*	*20*

ID	*log(k^0)[a]*	*D ($X10^{-5}$ cm^2 s^{-1})[a]*	*E(V)[b]*	*i_{appl} (mA cm^{-2})*	*c_{act} (mol L^{-1})*	*ω (mL min^{-1})*
10	*-2.3*	*0.25*	*0.8*	*80*	*1*	*5*
11	*-5.3*	*2.5*	*0.8*	*80*	*1*	*5*
12	*-2.3*	*2.5*	*0.8*	*80*	*1*	*20*
13	*-5.3*	*0.25*	*1.2*	*80*	*1*	*5*
14	*-2.3*	*0.25*	*1.2*	*80*	*1*	*20*
15	*-5.3*	*2.5*	*1.2*	*80*	*1*	*20*
16	*-2.3*	*2.5*	*1.2*	*80*	*1*	*5*
17	*-5.3*	*0.25*	*0.8*	*50*	*2*	*20*
18	*-2.3*	*0.25*	*0.8*	*50*	*2*	*5*
19	*-5.3*	*2.5*	*0.8*	*50*	*2*	*5*
20	*-2.3*	*2.5*	*0.8*	*50*	*2*	*20*
21	*-5.3*	*0.25*	*1.2*	*50*	*2*	*5*
22	*-2.3*	*0.25*	*1.2*	*50*	*2*	*20*
23	*-5.3*	*2.5*	*1.2*	*50*	*2*	*20*
24	*-2.3*	*2.5*	*1.2*	*50*	*2*	*5*
25	*-5.3*	*0.25*	*0.8*	*80*	*2*	*5*
26	*-2.3*	*0.25*	*0.8*	*80*	*2*	*20*
27	*-5.3*	*2.5*	*0.8*	*80*	*2*	*20*
28	*-2.3*	*2.5*	*0.8*	*80*	*2*	*5*
29	*-5.3*	*0.25*	*1.2*	*80*	*2*	*20*
30	*-2.3*	*0.25*	*1.2*	*80*	*2*	*5*
31	*-5.3*	*2.5*	*1.2*	*80*	*2*	*5*
32	*-2.3*	*2.5*	*1.2*	*80*	*2*	*20*

[a]For both anolyte and catholyte, [b]E^0(anolyte)=-E/2 and E^0(catholyte)=E/2.

Performance metrics. The performance metrics were calculated in the same way as described for Study 1. The only difference is the calculation of voltage efficiency:

$$VE = \frac{\bar{V}_d}{\bar{V}_c} = \frac{\left(\sum_0^{n_{SoC,d}} V_d / n_{SoC,d}\right)}{\left(\sum_0^{n_{SoC,c}} V_c / n_{SoC,c}\right)} \quad \textbf{(139)}$$

Where $n_{SoC,d}$ and $n_{SoC,c}$ are the number of calculations performed for each SoC condition in the discharging and charging procedure, respectively.

5 Understanding how operating conditions affect optimal geometry for vanadium redox flow batteries

This chapter shows the results of Study 1. The first section shows a brief introduction to the study. The second section describes how geometric modifications affect energy efficiency in VRFBs. The third section shows the figures of merit from regression analysis. The next section discusses the coefficients obtained from regression analysis and the consequences of their values. Section 5.4 describes a new method to calculate the optimal geometry for VRFBs operating at flow-through design. And section 5.5 brings the details about the validation of regression analysis.

5.1 Introduction to the chapter

Since geometric variables can affect the pumping energy consumption and the ohmic and concentration overpotentials, we carried out a study to investigate the effects of geometric modifications in VRFBs' performance. We innovated by (i) using a geometry larger than those in laboratory scale (400 cm^2 of area); (ii) using modifiable geometry; (iii) studying the interaction effects between geometric modification and operating conditions; (iv) coupling multi-physical model and regression analysis to deeply explore the system; (v) proposing a method to calculate the optimal geometry to any flow-design (flow-by or flow-through), type and scale of RFBs.

Figure 24 shows a graphical abstract for Study 1.

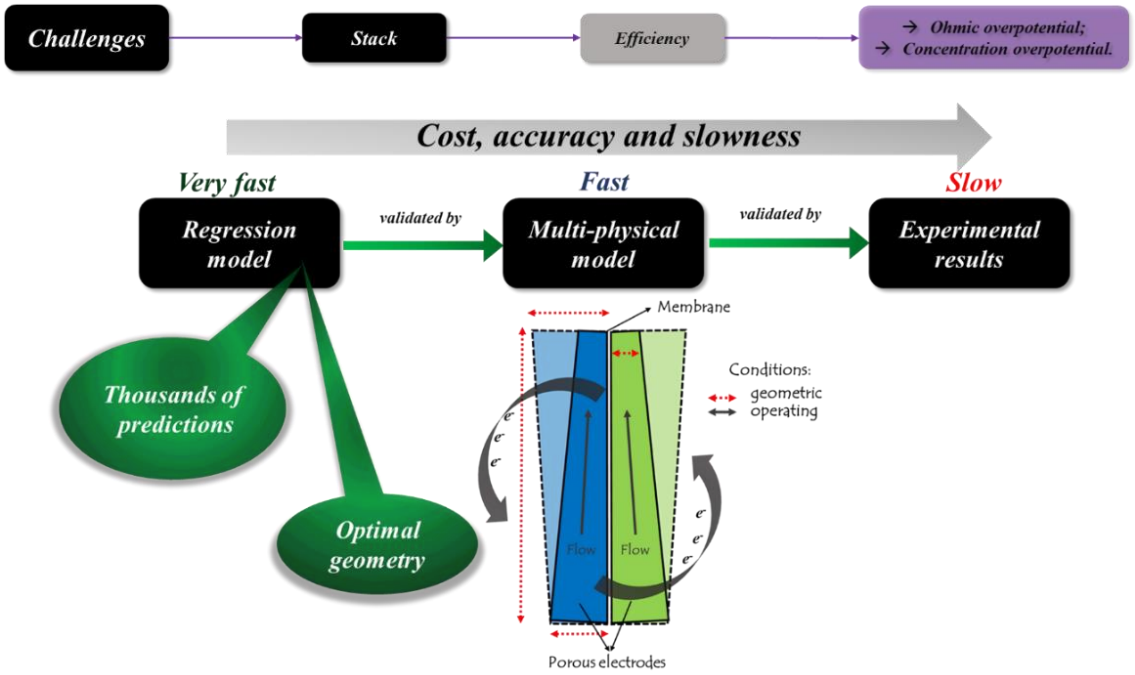


Figure 24. *Graphical abstract for Study 1.*

5.2 Understanding the geometric variables

Before applying the method of approach, we aimed to understand how each geometric condition affects the pumping energy consumption, the concentration overpotential, and ohmic overpotential. In the next pages, we discuss the effects of the length of residence path (*H*), the electrode thickness (*L*), and the trapezoidal geometry on the battery efficiency of VRFBs.

Length of residence path (H). The increase of the length of residence path leads to the decrease of regions of non-uniform concentration (see **Figure 25**). However, the decrease of these regions of non-uniform concentration does not influences the concentration overpotential

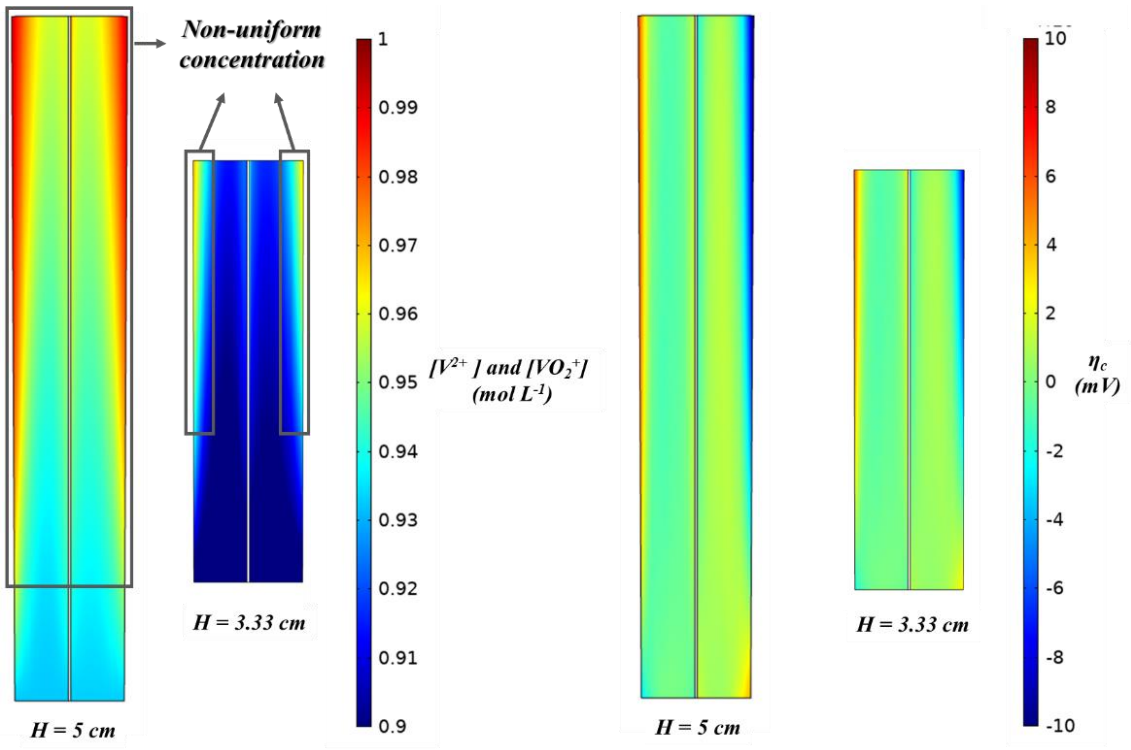


Figure 25. *Regions of non-uniform concentration (left) and the respective concentration overpotential (right) for two cells with different lengths of residence path in the discharging procedure. Initial concentration of active species is equal to 1 mol* L^{-1}.

Another characteristic of decreasing the length of the residence path is the decrease of power required for electrolyte recirculation. Thus, a smaller pump power is required and, as consequence, the pumping energy consumption by cycle decreases. A schematic representation of this conclusion is shown in **Figure 26**.

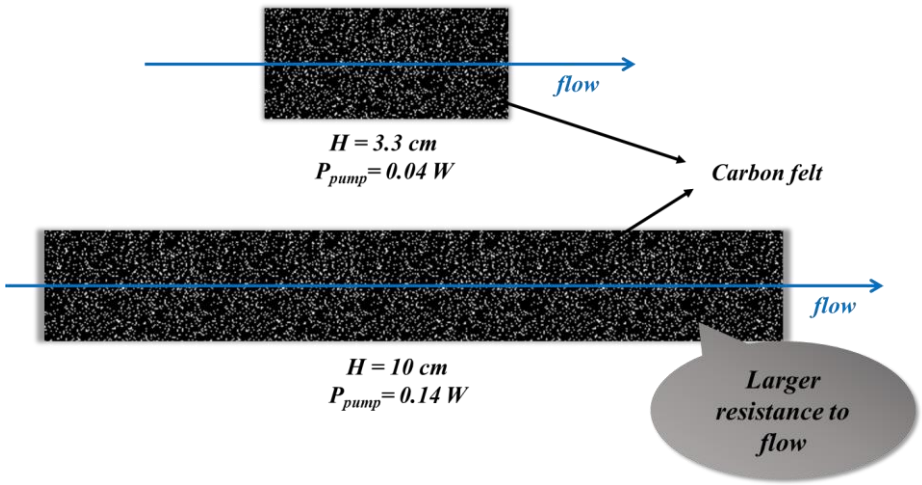


Figure 26. *Schematic representation comparing the effects of length of residence path on resistance to flow inside the half-cells of carbon felt.*

A larger length of residence path (10 cm) requires more power of the pump (0.14 W) to overcome the resistance to flow caused by the presence of carbon felt inside the cell. If the length of the residence path is smaller (3.3 cm), the power required for recirculation of electrolyte is also smaller (0.04 W), because the resistance to flow decreases. Now, it is necessary to evaluate if this consequence on pump power is sufficient to change the efficiency of the cell.

Electrode thickness (L). The increase of electrode thickness has the following consequences:

(i) Increase of cell electrical resistance. The increase of electrode thickness also increases the path taken by current and, as consequence, the ohmic overpotential increases (see **Figure 27**).

(ii) Decrease of pump power required for electrolyte circulation due to the decrease of resistance to flow caused by thicker channels. This condition also led to the decrease of flow velocity (v) inside the cell, considering the same volumetric flow (ω):

$$\downarrow \boldsymbol{v} = \frac{\omega}{\varepsilon w_{cell} \boldsymbol{L}\uparrow} \qquad \textbf{(140)}$$

If the flow velocity decrease, the mass transport is less effective and the concentration overpotential increases.

(iii) Increase of active area per length of residence path[105]. That is when thicker electrodes are used the active area available for the electrochemical reaction increase along the residence path, making the mass transport to the active surface easier (see **Figure 27**). This is because there is less traffic for the mass transport from the bulk to the electrode surface since there are more active sites available for the electrochemical reaction. Since the mass transport is more effective, the concentration overpotential is mitigated as the electrode is made thicker.

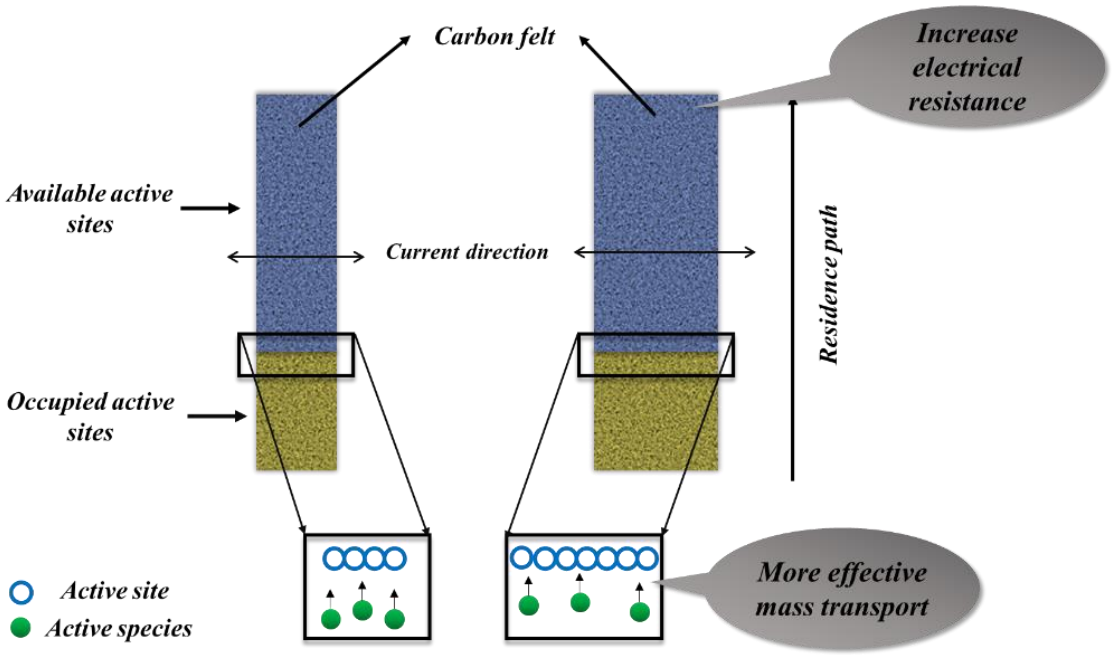
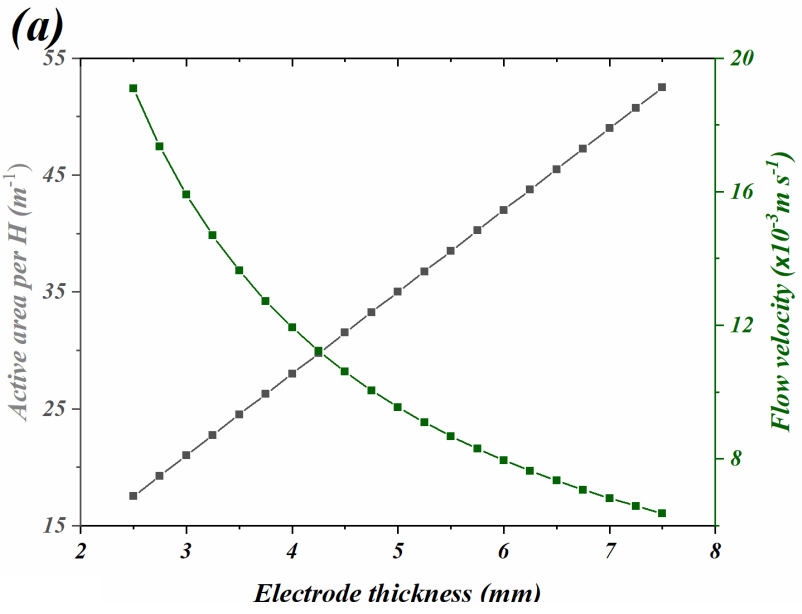


Figure 27. *Comparing the effects of electrode thickness in the mass transport.*

The effects (ii) and (iii) are antagonists because one increases the concentration overpotential towards thicker electrodes and the other decreases the concentration overpotential towards thicker electrodes. The net effect depends on the cell scale. For a cell with 400 cm^2 of area (w_{cell} X H), the more important effect is the increase of active are per length of residence path (iii). **Figure 28** shows the effect of increasing electrode thickness in the active area per length of residence path and flow velocity (a) and the net effect on the difference between bulk and surface concentration (b) for a 400 cm^2 cell.



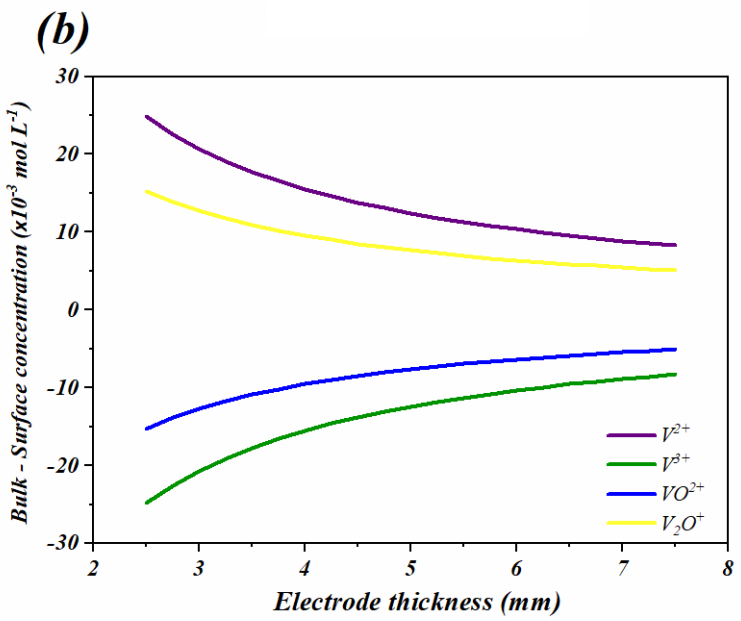


Figure 28. *The effect of electrode thickness in (a) the active area along the residence path and the flow velocity, and (b) in the difference between bulk and surface concentrations at discharging procedure.*

Figure 28(b) shows that the difference between bulk and surface concentration of vanadium species decreases as electrode thickness increases, proving that increase of active area per length of residence path (effect (iii)) is more important than the decrease of flow velocity (effect (ii)) in this scale. That way, the increase of electrode thickness mitigates the overpotential concentration.

Trapezoidal geometry. The trapezoidal geometry was already studied[66,67]. The purpose is to accelerate the electrolyte from the inlet to the outlet - without interfering in the operational conditions – to mitigate the regions of non-uniform concentration (see **Figure 29**). However, these studies modified the w_{cell}, without interference in electrode thickness. Our purpose is to add to this effect of electrolyte acceleration, the effects of electrode narrowing: modification of ohmic and concentration overpotentials.

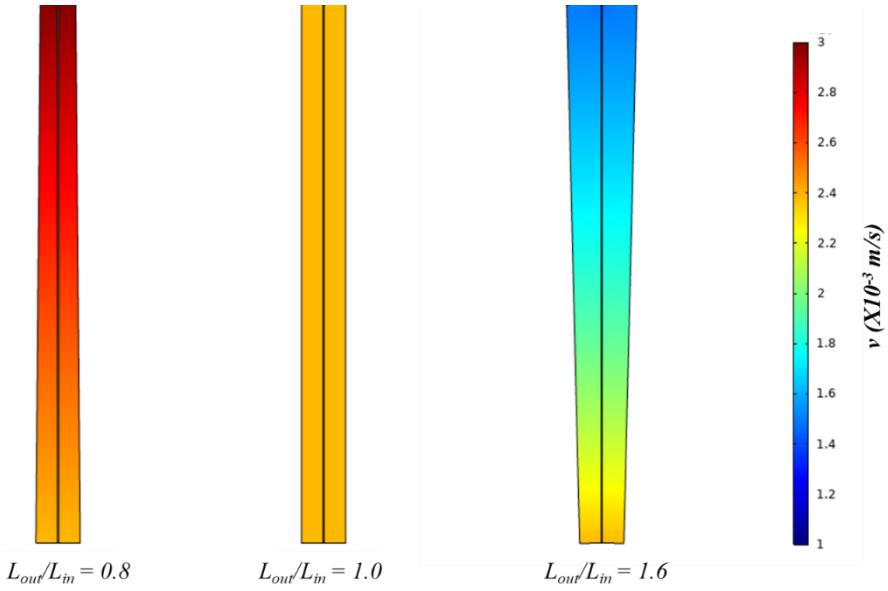


Figure 29. *Comparing the effects of L_{out}/L_{in} on flow velocity.*

In the first case, for $L_{out}/L_{in} = 0.8$, the geometry is trapezoidal and the flow velocity decreases along the residence path. In the second case, we have the typical rectangular geometry with the constant flow velocity. For $L_{out}/L_{in} = 1.6$, the geometric is an inverted trapezoid and the velocity decreases along the residence path.

Beyond the effects on the velocity flow, the trapezoid geometries influence the electrical resistance and the active area per unit of H. Thus, for $L_{out}/L_{in} = 0.8$, the ohmic resistance decreases, and the concentration overpotential increases along the residence path. And, for $L_{out}/L_{in} = 1.6$, the ohmic resistance increases, and the concentration overpotential decreases along the residence path.

Next steps. The next section discusses the effect of these three geometric variables with two operational conditions (applied current density and volumetric flow) on the efficiency of the battery.

5.3 Regression analysis

Five responses related to performance were evaluated: voltage efficiency (*VE*), battery efficiency (*BE*), pumping energy consumption (E_{pump}), ohmic overpotential (η_o), and concentration overpotential (η_c). The complete list of results from designed

computational experiments (from designed experiments of **Table 14** in Methodology) is described in **Table 22**.

Table 22. *All responses from Doehlert design (Study 1)**

ID	*VE (%)*	*BE (%)*	*E_{pump} (J)*	*η_o (mV)*	*η_c (mV)*
1	*65.6*	*65.6*	*289.87*	*92.67*	*88.95*
2	*64.4*	*64.2*	*204.11*	*123.1*	*62.47*
3	*48.4*	*48.3*	*134.03*	*192.13*	*117.15*
4	*59.3*	*59.1*	*544.01*	*135.94*	*89.34*
5	*57.5*	*57.4*	*221.38*	*158.68*	*74.49*
6	*58.7*	*58.7*	*576.46*	*140.5*	*86.04*
7	*63.8*	*63.6*	*569.42*	*62.25*	*148.29*
8	*88.9*	*88.7*	*1718.95*	*17.08*	*28.85*
9	*70.6*	*70.4*	*47.77*	*57.4*	*88.86*
10	*70.4*	*70.4*	*459.13*	*49.23*	*108.87*
11	*88.8*	*88.8*	*286.65*	*16.6*	*30.28*
12	*72.3*	*72.2*	*322.71*	*79.82*	*56.11*
13	*71.4*	*71.2*	*30.08*	*79.95*	*56.76*
14	*72.0*	*72.0*	*287.18*	*66.41*	*71.53*
15	*72.2*	*72.1*	*53.70*	*77.16*	*58.68*
16	*51.2*	*51.1*	*212.20*	*137.95*	*165.75*
17	*52.4*	*52.0*	*17.75*	*141.35*	*130.25*
18	*55.2*	*55.1*	*170.11*	*118.89*	*148.41*
19	*54.8*	*54.7*	*31.82*	*136.54*	*127.65*
20	*60.1*	*59.9*	*863.60*	*97.59*	*131.53*
21	*80.3*	*80.0*	*1735.58*	*44.52*	*46.24*
22	*65.6*	*65.5*	*880.48*	*78.09*	*109.89*
23	*65.9*	*65.8*	*163.89*	*89.78*	*92.33*
24	*60.0*	*59.9*	*348.11*	*111.25*	*112.8*
25	*80.4*	*80.3*	*704.57*	*51.45*	*37.73*
26	*65.0*	*64.8*	*56.14*	*107.25*	*74.56*
27	*65.6*	*65.5*	*58.97*	*104.23*	*76.65*
28	*60.1*	*59.8*	*914.79*	*100.35*	*127.5*
29	*80.3*	*80.0*	*1839.64*	*45.91*	*44.29*
30	*64.9*	*64.8*	*147.18*	*95.67*	*85.99*
31	*65.7*	*65.5*	*815.70*	*81.07*	*105.14*

**Algorithm developed to calculate responses is in Appendix A*

The pumping energy consumption was irrelevant to a considerable lowering of the battery efficiency. This was evaluated by the absolute difference between VE and BE for all computational experiments, which are in the same scale of RMSE (see **Figure 30**). This means that there was no statistical difference between BE and VE. Thus, the evaluation of BE is redundant. This also indicates that the pumping energy consumption

(E_{pump}) is low to reduce the energy efficiency in the studied scale. Thus, E_{pump} is not important in being studied too.

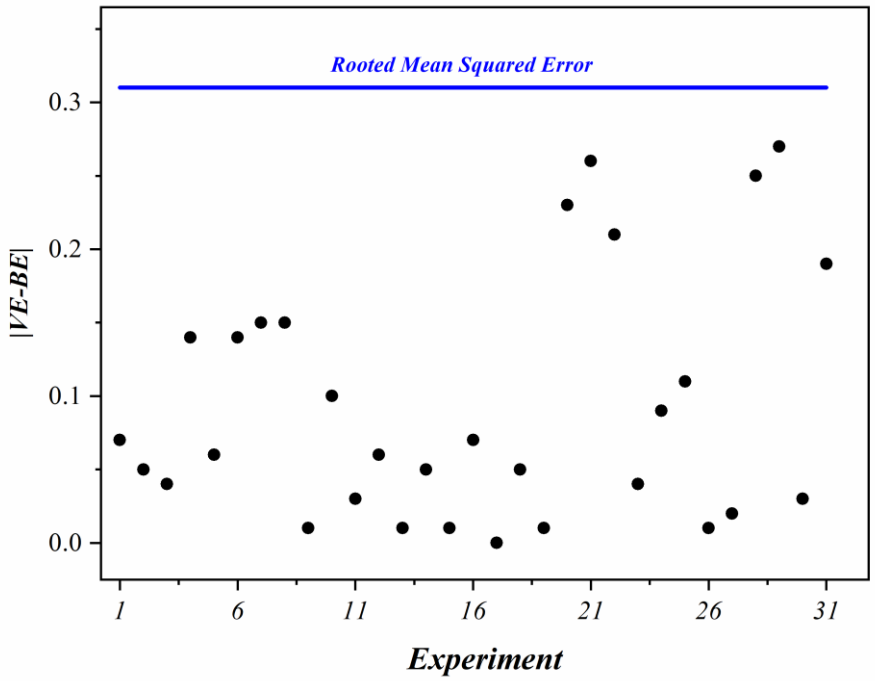


Figure 30. *A plot of the absolute difference between calculated voltage and battery efficiencies for each computational experiment, evidencing that there is no statistical difference between these two responses.*

The remaining responses are *VE,* η_o, and η_c, which were submitted to regression analysis. **Table 23** shows the figures of merit for regression analysis.

Table 23. *Figures of merit for regression analysis**

	VE (%)	η_o *(mV)*	η_c *(mV)*
R^2	*0.9991*	*0.9999*	*0.9977*
RMSE	*0.31*	*0.30*	*1.74*

**Algorithm developed for regression analysis is in Appendix A*

The coefficients of determination, R^2, indicated that all responses were well-fitted to quadratic regression analysis. The responses also have low values of root mean squared error (RMSE). These conditions allowed the use of the regression models to a robust prediction of *VE,* η_o, and η_c in the experimental space.

Empirical equations. The empirical equations for *VE,* η_o and η_c, considering only the statistically significant coefficients at $p < 0.05$ level based on RMSE, are, respectively:

$$VE = 65.43 - 22.57i_{appl} + 0.89\omega - 1.39L_{in}^2 - 2.23L_{in}i_{appl} - 1.69L_{in}(L_{out}/L_{in}) + 6.19{i_{appl}}^2 + 1.24i_{appl}\omega - 1.59i_{appl}(L_{out}/L_{in}) - 0.82\omega^2 \quad \textbf{(141)}$$

$$\eta_o = 93.03 + 31.23L_{in} + 83.47i_{appl} + 1.27H + 18.48\,(L_{out}/L_{in}) + 26.94L_{in}i_{appl} + 9.02L_{in}(L_{out}/L_{in}) + 3.45i_{appl}H + 15.92i_{appl}(L_{out}/L_{in}) \quad \textbf{(142)}$$

$$\eta_c = 89.62 - 39.07L_{in} + 69.68i_{appl} - 22.25\,(L_{out}/L_{in}) + 15.81L_{in}^2 - 15.25L_{in}i_{appl} + 5.26L_{in}(L_{out}/L_{in}) - 18.47{i_{appl}}^2 - 7.32i_{appl}\omega - 5.43i_{appl}H - 12.28i_{appl}(L_{out}/L_{in}) + 4.44(L_{out}/L_{in})^2 \quad \textbf{(143)}$$

The coefficients are calculated from the codified variables.

We change the variable *L*, discussed in the previous section, by L_{in}. This change in nomenclature is necessary when we are working with the variable L_{out}/L_{in} at the same time. Thus, we can change only L_{in}, because the electrode thickness (*L*) changes along the residence path.

The next sections discuss the coefficients of the above equations and their consequences on the performance of VRFBs.

5.3.1 Evaluation of linear and quadratic coefficients

Table 24 shows the statistically significant "pure" (linear and quadratic) coefficients from equations above of *VE,* η_o, and η_c. The term pure is used because the linear and quadratic coefficients are calculated separately for each variable, differently from the interaction coefficients, which is going to be discussed in section 5.2.2.

Table 24. *Calculated linear and quadratic coefficients*,[a]*

		Response		
Variable	*Coefficients*	η_o *(mV)*	η_c *(mV)*	*VE (%)*
L_{in}	w_{01}	*+31.23*	*-39.07*	-
L_{in}^2	w_{11}	-	*+15.81*	*-1.39*
i_{appl}	w_{02}	*+83.47*	*+69.68*	*-22.57*
${i_{appl}}^2$	w_{22}	-	*-18.47*	*+6.19*
ω	w_{03}	-	-	*+0.89*
ω^2	w_{33}	-	-	*-0.82*
H	w_{04}	*+1.27*	-	-
H^2	w_{44}	-	-	-
L_{out}/L_{in}	w_{05}	*+18.48*	*-22.25*	-
$(L_{out}/L_{in})^2$	w_{55}	-	*+4.44*	-

**Algorithm developed to determine the statistical significance is in Appendix A*
[a]The non-significant coefficients are omitted for clarity

The use of codified values is important because sets all variables and coefficients on the same scale. Another advantage of using codified variables (from -1 to +1) is that linear and quadratic coefficients can be compared in magnitude. For example, if both linear and quadratic coefficients are equal to 1, the partial response corresponding to each one of the coefficients assumes the behavior portrayed in **Figure 31**.

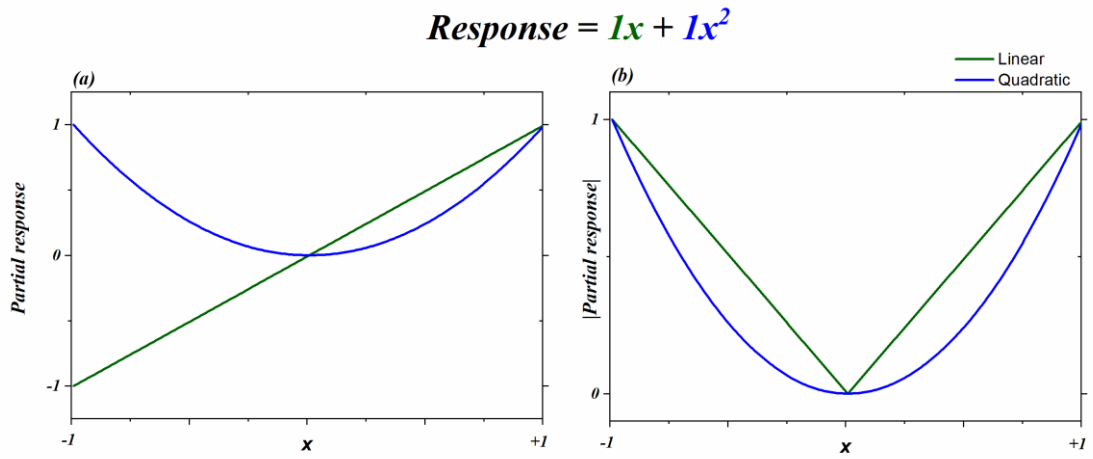


Figure 31. *Comparison between linear and quadratic partial response (a) and absolute partial response (b).*

Despite the linear and quadratic partial responses being very different (**Figure 31**(a)), their absolute values are very similar (**Figure 31**(b)), which makes possible the comparison between the coefficients. Considering this characteristic, in the next pages, a comparison between the "pure" coefficients is made, and the physical consequences of their values are discussed.

Coefficients of inlet electrode thickness (L_{in}) (w_{01} and w_{11}). For *ohmic overpotential*, just the linear coefficient was statistically significant ($w_{01}(\eta_o)=+31.23$), since the increase of electrode thickness increases the cell electrical resistance, as discussed in section 5.1.

For *concentration overpotential*, the magnitude of the linear coefficient is more than double of quadratic coefficient: $w_{01}(\eta_c)=-39.07$ *and* $w_{11}(\eta_c)=+15.81$, indicating the prevalence of a linear behavior instead of a quadratic one (see **Figure 32**).

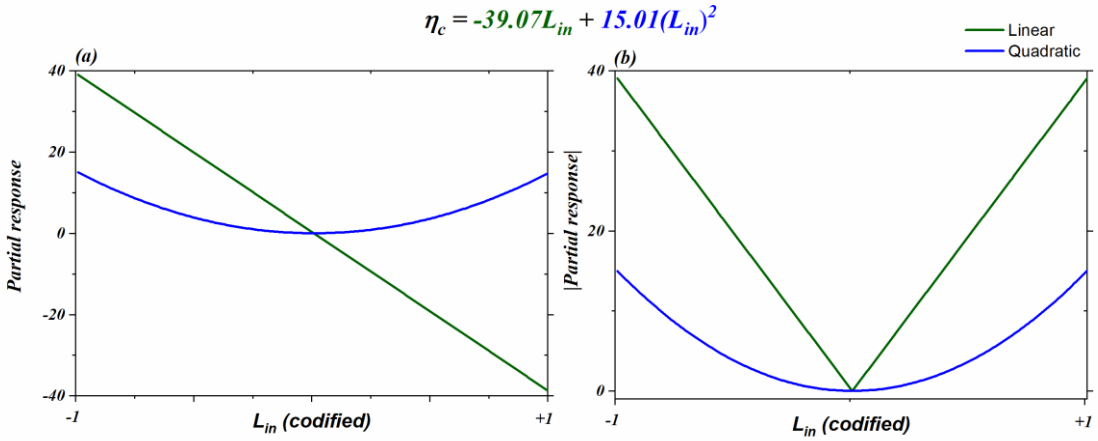


Figure 32. *Comparison between linear and quadratic partial response (a) and absolute partial response (b) for concentration overpotential concerning inlet electrode thickness.*

The physical meaning of this is related to the more important effect of increasing of active area per length of residence path as electrode thickness increases (iii) than the effect of decreasing flow velocity in this same direction (ii), as discussed in Section 5.1. Thus, despite there being two antagonistic forces for decreasing the concentration overpotential, one is more important than the other, making the mitigation of overpotential possible as electrode thickness increases. **Figure 33** illustrates this conclusion by showing the partial responses and the total response for concentration overpotential concerning the inlet electrode thickness.

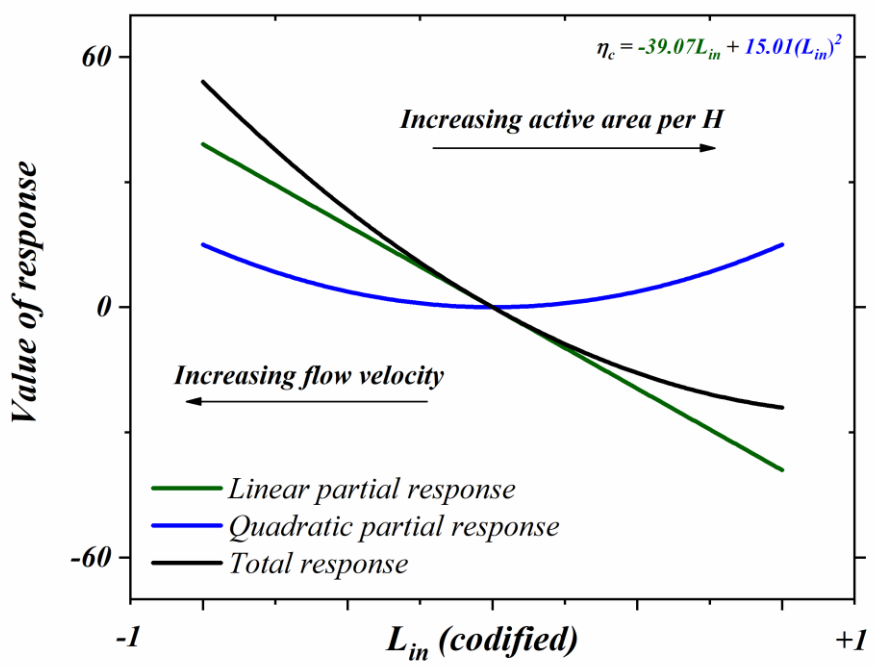


Figure 33. *The linear and quadratic partial responses and the total response for the concentration overpotential concerning inlet electrode thickness.*

Finally, the opposite effects for ohmic (increases as electrode thickness increases) and concentration overpotential (decreases as electrode thickness decreases) explain the statistical significance of the quadratic coefficient for *voltage efficiency* $w_{11}(VE) = -1.39$, where a maximum response is expected.

Coefficients of applied current density (i_{appl}) (w_{02} and w_{22}). For *ohmic overpotential*, just the linear coefficient was statistically significant ($w_{02}(\eta_o)=+87.43$), since the increase of current increases the ohmic overpotential by definition.

For *concentration overpotential*, the magnitude of the linear coefficient is more than triple of quadratic coefficient: $w_{02}(\eta_c)=-69.68$ *and* $w_{22}(\eta_c)=-18.47$, indicating the prevalence of a linear behavior instead of a quadratic one. In the same way, we did for inlet electrode thickness, we can visualize this mathematical behavior plotting the partial and total responses (see **Figure 34**).

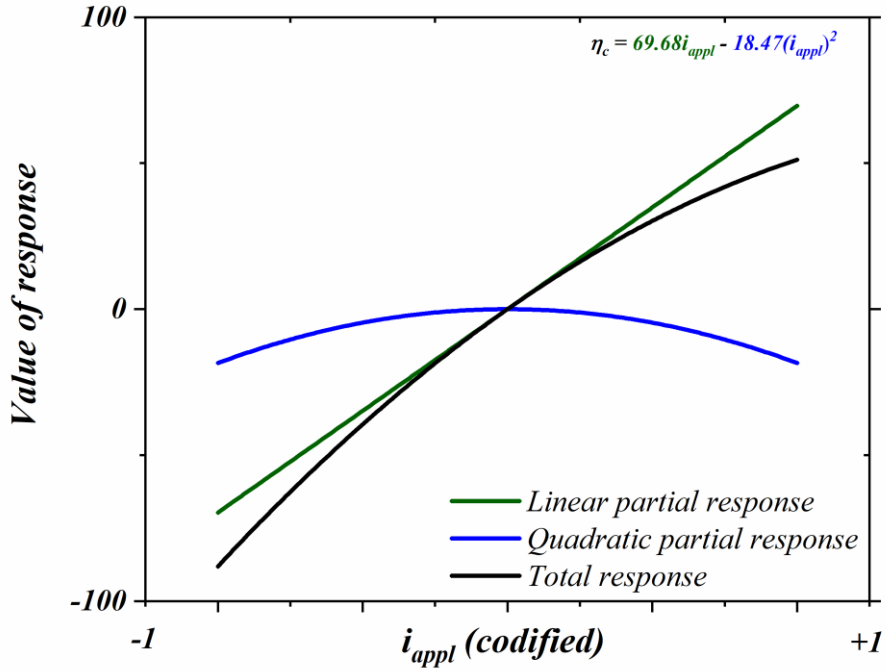


Figure 34. *The linear and quadratic partial responses and the total response for the concentration overpotential concerning applied current density.*

The physical meaning for this correlation between concentration overpotential and applied current density is related to the demand for efficient mass transfer. High applied current densities lead to faster reactions and the active species at the electrode surface need to be replaced faster too. When this demand for mass transfer is not met, large concentration overpotential arises.

The high positive linear correlation between applied current density and both concentration overpotential and ohmic overpotentials implies a large negative linear coefficient for *voltage efficiency* ($w_{02}(VE)=-22.57$). Thus, as expected, the increasing of applied current density leads to lower voltage efficiencies.

Coefficients of flow velocity (ω) (w_{03} and w_{33}). The flow velocity does not affect ohmic overpotential. And the effect on concentration overpotential was insignificant, indicating that the range of values of flow velocity employed was sufficient for the mitigation of concentration overpotential.

Coefficients of the length of residence path (H) (w_{04} and w_{44}). The only significant coefficient for this variable was the linear one for *ohmic overpotential* ($w_{04}(\eta_o)=+1.27$) , but this value has no physical meaning. A variable just affects ohmic overpotential by increasing the cross-sectional area of the cell or by increasing the electrode thickness. The

length of the residence path does not affect the electrical resistance because the cross-sectional area of the cell (400 cm^2) is kept fixed in all experiments and has no effect on electrode thickness. Graphs of ohmic overpotential versus applied current densities for different values of *H* are shown in **Figure 35**, proving the no modification in the ohmic overpotential caused by *H*.

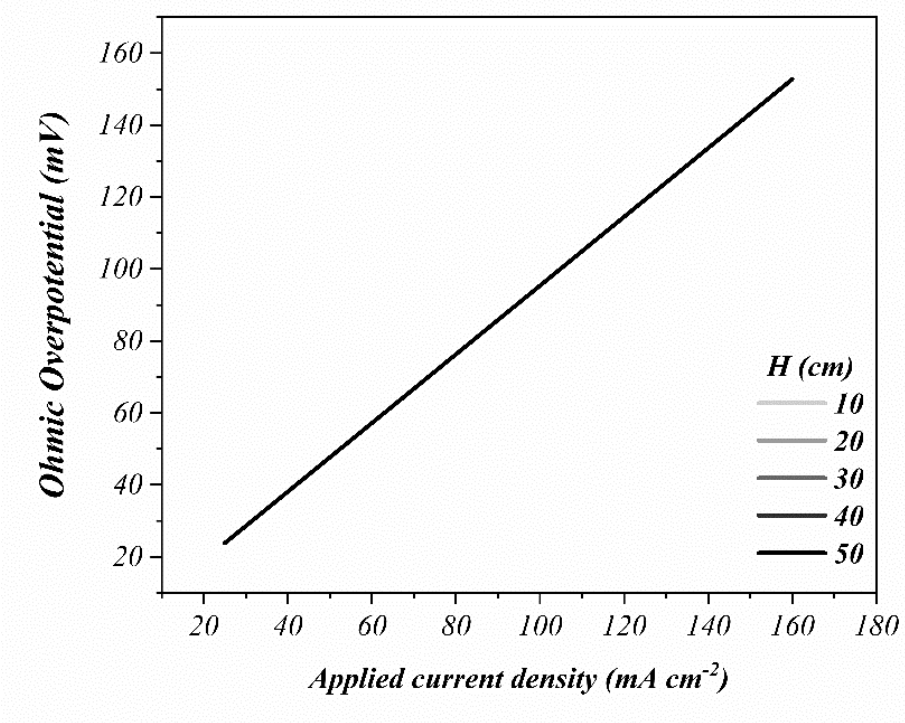


Figure 35. *Ohmic overpotential versus applied current density for different length of residence path*

Thus, despite regression analysis being a good tool for system understanding, its main purpose is fitting. Because of that, the analysis of the significance for the coefficients has to be made in both statistical and physical approaches.

Coefficients of L_{out}/L_{in} (w_{05} and w_{55}). Increasing L_{out}/L_{in} implies in (i) the increasing of *ohmic overpotential* ($w_{05}(\eta_o) = +18.48$) and (ii) the decrease of *concentration overpotential* ($w_{05}(\eta_c) = -22.55$). These are the same qualitative effects caused by the modification of inlet electrode thickness. However, the coefficients related to L_{out}/L_{in} for *voltage efficiency* (w_{05}*(VE)* and w_{55}*(VE)*) were not statistically significant. That is, the effects on ohmic and concentration overpotential caused by L_{out}/L_{in} tend to offset each other.

Important variables. Therefore, the most important geometric variable for mitigation of overpotential was the inlet electrode thickness. The length of the residence path and the L_{out}/L_{in} had no net effect on voltage efficiency. For operating conditions, the applied current density was relevant and the volumetric flow was sufficient for the mitigation of concentration overpotential.

5.3.2 Evaluation of interaction effects

The analysis performed in the previous section considered just linear and quadratic coefficients. However, interaction effects also contribute to battery performance. Table **Table 25** shows the statistically significant (at the $p < 0.05$ level based on RMSE) interaction effects.

Table 25. *Calculated interaction coefficients*[*,a]*

			Response		
Variable 1	*Variable 2*	*Coefficients*	η_o	η_c	*VE*
$\boldsymbol{L}$	$\boldsymbol{i_{appl}}$	$\boldsymbol{w_{12}}$	***+26.94***	***-15.25***	***-2.23***
L	ω	w_{13}	-	-	-
L	H	w_{14}	-	-	-
$\boldsymbol{L}$	$\boldsymbol{L_{out}/L_{in}}$	$\boldsymbol{w_{15}}$	***+9.02***	***+5.26***	***-1.69***
$\boldsymbol{i_{appl}}$	$\boldsymbol{\omega}$	$\boldsymbol{w_{23}}$	-	***-7.32***	***+1.24***
i_{appl}	H	w_{24}	***+3.45***	***-5.43***	-
$\boldsymbol{i_{appl}}$	$\boldsymbol{L_{out}/L_{in}}$	$\boldsymbol{w_{25}}$	***+15.92***	***-12.28***	***-1.59***
ω	H	w_{34}	-	-	-
ω	L_{out}/L_{in}	w_{35}	-	-	-
H	L_{out}/L_{in}	w_{45}	-	-	-

*Algorithm developed to determine the statistical significance is in Appendix A

[a]The non-significant coefficients are omitted for clarity

Negative effects between geometric parameters and applied current density (w_{12} and w_{25}). The statistically significant interaction coefficients between geometric parameters and applied current density were w_{12} for inlet electrode thickness and w_{25} for L_{out}/L_{in}. Despite L_{out}/L_{in} is irrelevant by itself for voltage efficiency, its interaction with other variables contributes to cell performance. This is a finding that univariate analysis usually neglects and is very common in electrochemical systems. These two negative significant coefficients indicate that when large current densities are applied for thick

electrodes (high values of L_{in} and/or L_{out}/L_{in}) there is an additional contribution to the decrease of voltage efficiency ($w_{12}(VE) = -2.23$ and $w_{25}(VE) = -1.59$).

These negative effects on voltage efficiency are caused by a higher coefficient for ohmic overpotential than for concentration overpotential. For L_{in}, the coefficient of ohmic overpotential is almost double of the coefficient for concentration overpotential: $w_{12}(\eta_o) = +26.94$ *and* $w_{12}(\eta_c) = -15.25$. For L_{in}/L_{out}, the coefficient of ohmic overpotential is close to the coefficient for concentration overpotential: $w_{25}(\eta_o) = +15.92$ *and* $w_{25}(\eta_c) = -12.28$.

The larger contribution for ohmic overpotential than for concentration overpotential, when large currents and thick electrodes are applied (high values of L_{in} and/or L_{out}/L_{in}), is physically explained by the definition of ohmic overpotential itself:

$$(\eta_o)_l^{e,j} = i_{appl} \frac{L_{avg}}{\varepsilon^{3/2}\sigma_l^{e,j}} \qquad \textbf{(144)}$$

Where L_{avg} is the average of electrode thickness, which increases by increasing both L_{in} and L_{out}/L_{in}.

That is, the increasing of i_{appl}, L_{in} and L_{out}/L_{in} increases the ohmic overpotential. Thus, there is an interaction effect between these variables by definition, explaining the positive values of $w_{12}(\eta_o) = +26.94$ and $w_{25}(\eta_o) = +15.92$.

This synergetic effect in ohmic overpotential does not happen for the concentration overpotential, since high applied current densities increase the concentration overpotential, whereas thick electrodes decrease the concentration overpotential. That is an antagonistic effect, explaining the negative effects of $w_{12}(\eta_c) = -15.25$ and $w_{25}(\eta_c) = -12.28$.

Because of this, the interaction effect of ohmic overpotential is more relevant, and the voltage efficiency tends to be lower when applied current densities are used with thick electrodes. **Figure 36** schematizes these considerations.

In the first case, thin electrodes are operating in small currents. The ohmic overpotential is low because of both small current and thin electrodes (blue arrows at left). On the other hand, the concentration overpotential tends to be large because of the thin electrode at the same time that tends to be small because of the small current (red arrows at left).

In the second case, the thick electrodes are operating in large currents. The ohmic overpotential is high because of both small current and thin electrodes (blue arrows at right). On the other hand, the concentration overpotential tends to be small because of the thick electrode at the same time that tends to be large because of the large current (red arrows at right).

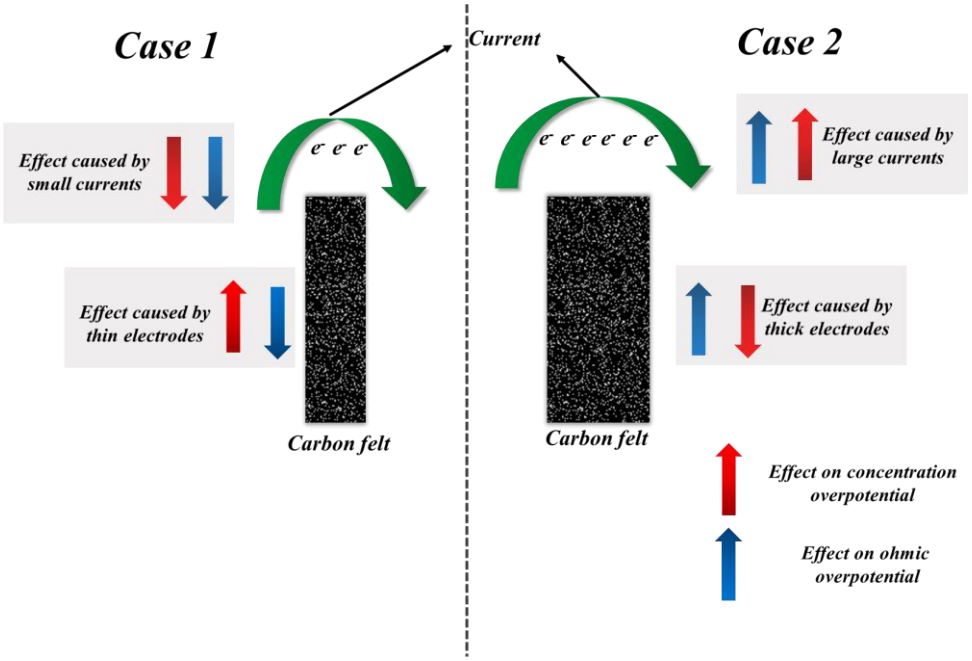


Figure 36. *Schematic representation comparing thin electrodes operating in small currents with thick electrodes operating in large currents.*

Negative effects between geometric parameters (w_{15}). Another significant interaction arises from electrode thickness with L_{out}/L_{in} on voltage efficiency ($w_{15}(VE)$=-*1.69*). This is because both variables increase the average electrode thickness, leading to large ohmic overpotentials and smaller concentration overpotentials.

Positive effects between operating conditions (w_{23}). An interaction effect arises from applied current density with the volumetric flow ($w_{23}(VE)=1.24$). This effect is associated with the demands for efficient mass transfer when applied current density increases. Thus, high applied current densities demand high flow velocity for the effective mitigation of concentration overpotential.

The effect between applied current density and length of residence path (w_{24}). This coefficient is statistically significant just for concentration overpotential and ohmic

overpotential. Despite statistical significance, the interaction effects for concentration and ohmic overpotential have no physical meaning. This is reinforced by curves of overpotential versus applied current density (see **Figure 35** and **Figure 37**), where no change was observed from the modification of the length of the residence path.

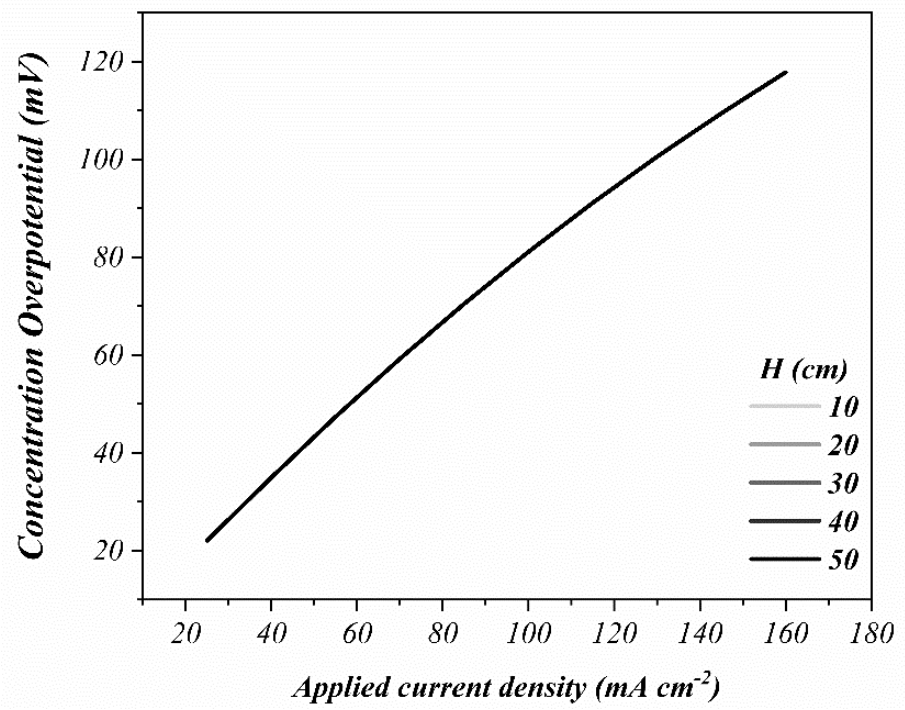


Figure 37. *Concentration overpotential versus applied current density for different lengths of residence path.*

Important effects. Therefore, the most relevant interactions were between L_{in}, L_{out}/L_{in}, and applied current density. The L_{out}/L_{in} ratio arises as a second geometric variable for mitigation of both ohmic and concentration overpotential.

5.3.3 Interaction effects between geometric parameters and applied current density

By analysis of the coefficients, we confirmed that interaction effects exist and are relevant for voltage efficiency. To further evaluate the relevant effects, three contour plots of L_{in} versus L_{out}/L_{in} for different applied current densities were stacked in a 3D graph (see **Figure 38**). The values of VE were normalized by the maximum value in each applied current density, since the values of vary greatly when different i_{appl} are used.

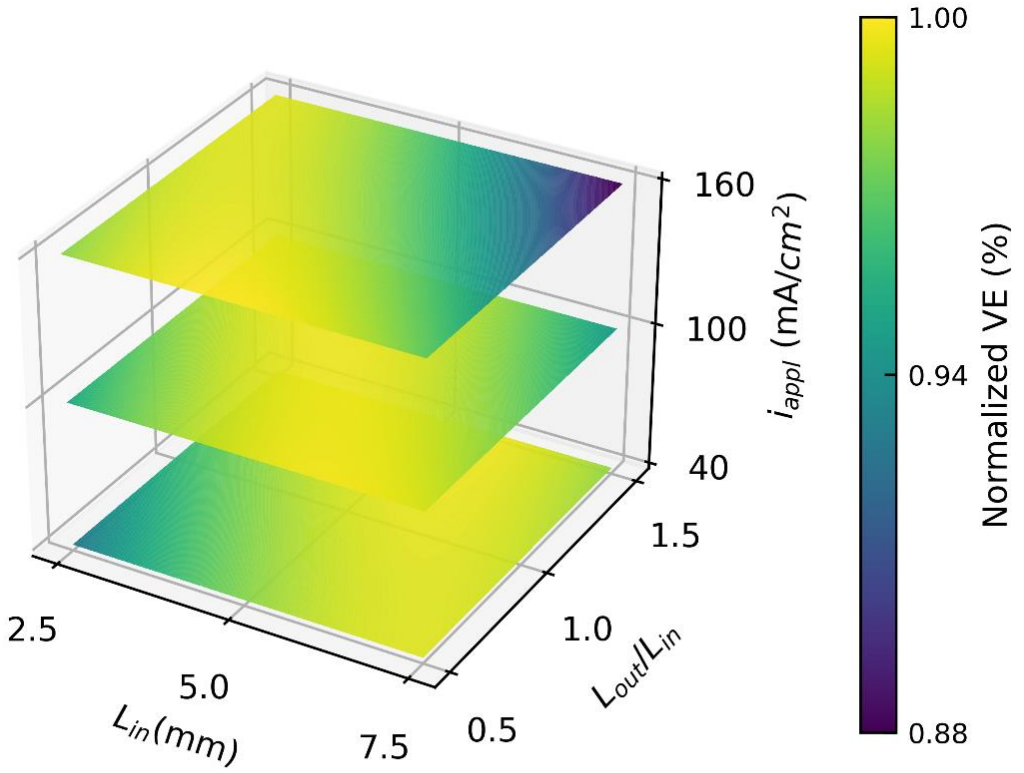


***Figure 38**. Contour plots of normalized voltage efficiency (VE) for inlet electrode thickness (L_{in}) versus the ratio between outlet electrode thickness and electrode thickness (L_{out}/L_{in}) for three different values of applied current density (i_{appl})*.*
**Algorithm developed to plot this graph is present in Appendix A.*

The predicted values indicated the inversion in the region of *VE* when the applied current density goes from 40 to 160 mA cm^{-2}. For large applied current densities (160 mA cm^{-2}; contour plot above), thin electrodes lead to optimum *VE*, whereas for smaller applied current densities (40 mA cm^{-2}; contour plot below) thick electrodes imply better voltage efficiency.

The above implications are explained in terms of the dominant overpotential in each condition. For small applied current densities (e.g., 40 mA cm^{-2}), the concentration overpotential is dominant concerning the ohmic overpotential (see **Figure 39**(a-b)). Thus, the voltage efficiency is benefited in direction of smaller concentration overpotentials (when the electrode is thick). On the other hand, the ohmic overpotential is dominant concerning the concentration overpotential for large applied current densities (see **Figure 39** (c-d)). Thus, the voltage efficiency is optimized in direction of small ohmic overpotential (when the electrode is thin). For intermediate cases (e.g. 100 mA cm^{-2}) there was a trade-off between the ohmic and concentration overpotential.

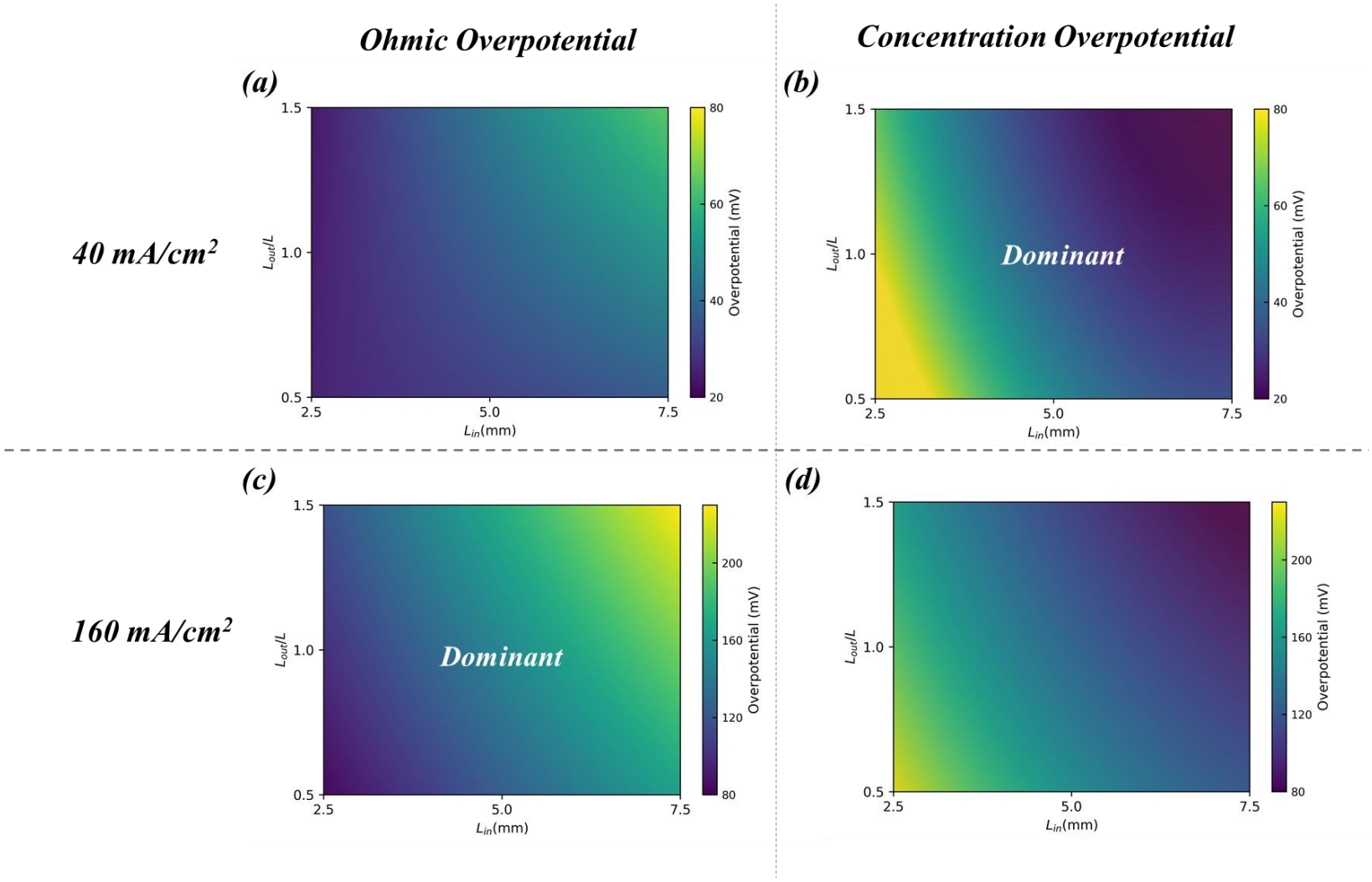


Figure 39. *Contour plots of concentration and ohmic overpotentials for electrode thickness (L) versus L_{out}/L_{in} for two different values of applied current density*.*
**Algorithm developed to plot this graph is in Appendix A.*

To further understand how to achieve maximum voltage efficiency for each applied current density, we calculated the optimal geometry. **Figure 40** shows these results.

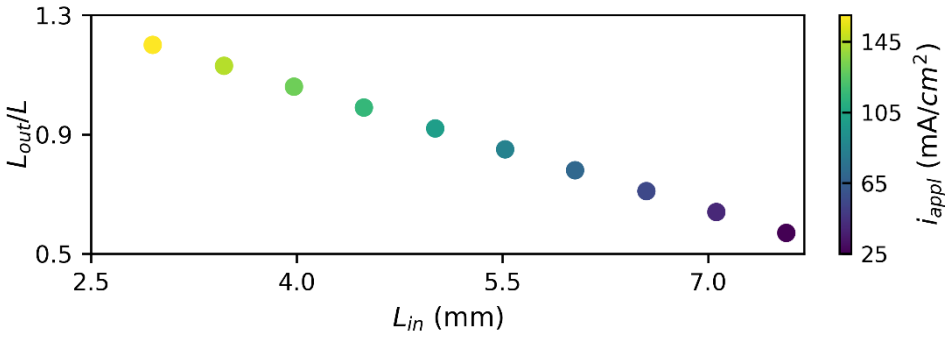


***Figure 40**. Optimum geometric conditions for each applied current density*.*
**Algorithm developed to plot this graph is in Appendix A.*

Since the dominant overpotential changes from concentration to ohmic when applied current density becomes large, the optimal inlet electrode thickness decreases in this direction (see Cases 1 and 2 in **Figure 41**).

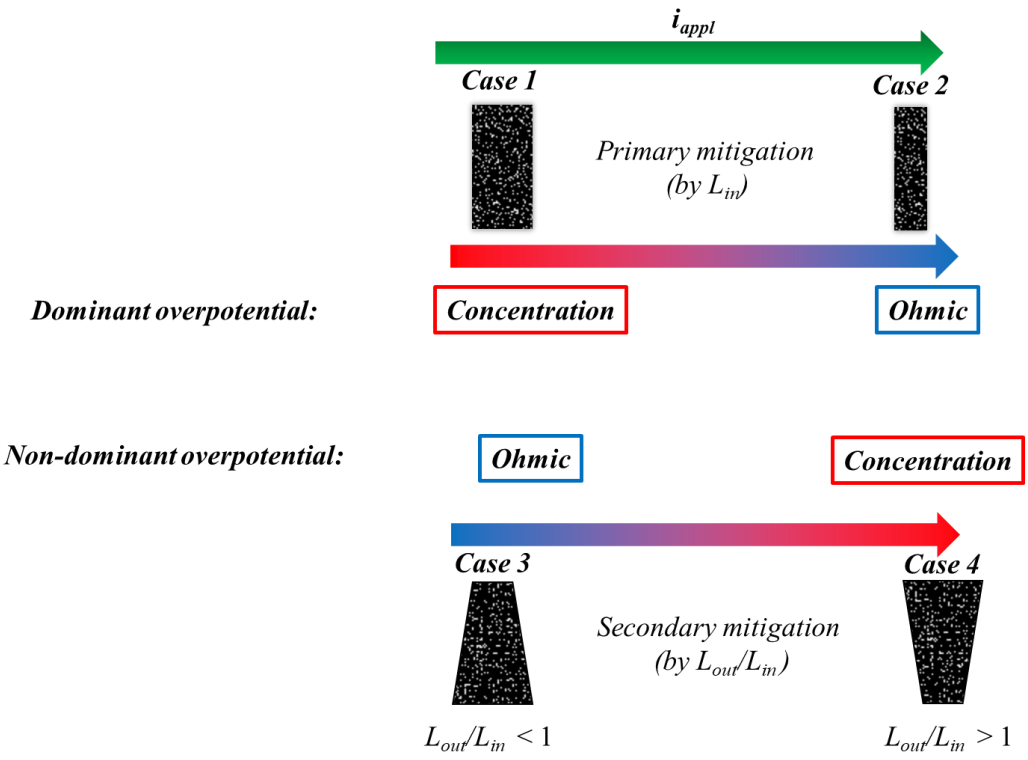


***Figure 41.** Schematic representation of dominant and non-dominant overpotential for a range of applied current density and the primary (by L_{in}) and secondary (by L_{out}/L_{in}) mitigations.*

Due to the focus on dominant overpotential, the modification of inlet electrode thickness is characterized as primary mitigation. For small applied current densities, thick electrodes mitigate the concentration overpotential (Case 1). On the other hand, thin electrodes mitigate the ohmic overpotential for large applied current densities (Case 2).

The voltage efficiency is also optimized by the enlargement of the electrode in outlet channels ($L_{out}/L_{in} > 1$) as the applied current increases. This characterizes a secondary mitigation strategy. Whereas the dominant overpotential is mitigated by electrode thickness, L_{out}/L_{in} modification is employed for the non-dominant overpotential. Therefore, small applied current densities require $L_{out}/L_{in} < 1$ to mitigate the ohmic overpotential (Case 3) and large applied currents densities require $L_{out}/L_{in} > 1$ to mitigate the concentration overpotential (Case 4).

Based on these results, we can calculate the percentage of mitigation for each condition. The percentage mitigation can be defined as the ratio of the overpotential in a given geometric condition, $\eta(L_{in},L_{out}/L_{in})$, by the maximum overpotential (η_{max}) for each applied current density:

$$Percentage\ of\ mitigation = \frac{\eta(L_{in},L_{out}/L_{in})}{\eta_{max}} \times 100\%^{*} \qquad \textbf{(145)}$$

*Algorithm developed to calculate the percentage of mitigation is in Appendix A.

The percentage of mitigation versus applied current density proves the aforementioned findings (see **Figure 42**). The percentage of mitigation for primary mitigation corresponds to the overpotential as a function of L_{in} for both ohmic and concentration overpotentials; that is $\eta_o(L_{in})$ and $\eta_c(L_{in})$, respectively. And the percentage mitigation for secondary mitigation corresponds to the overpotential as a function of L_{out}/L_{in} for both ohmic and concentration overpotentials; that is $\eta_o(L_{out}/L_{in})$ and $\eta_c(L_{out}/L_{in})$, respectively.

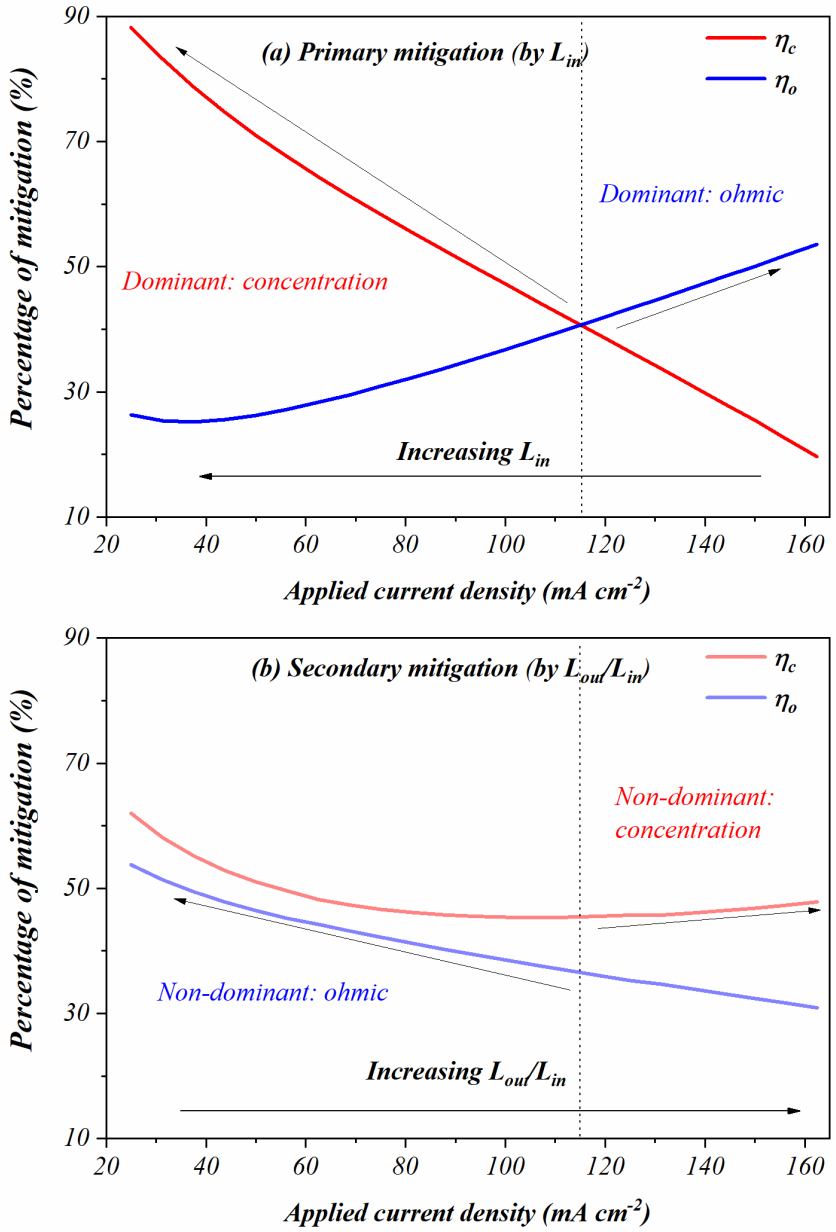


Figure 42. *Percentage mitigation of concentration overpotential (η_c) and ohmic overpotential (η_o) versus applied current density for (a) primary and (b) secondary mitigations.*

For the primary mitigation, there is a region of transition (close to 120 mA cm^{-2}) between ohmic overpotential and concentration overpotential. This is in agreement with (i) ohmic overpotential is dominant for large applied current densities and (ii) concentration overpotential is dominant for small applied current densities. **Figure 42** reinforces that secondary mitigation acts as a mitigation strategy for the non-dominant overpotential because the secondary mitigation increases as the primary mitigation decrease.

The first case occurs when the primary mitigation is directed to ohmic overpotential and the secondary mitigation is directed to concentration overpotential. In this case, the primary mitigation is so effective for small applied current densities that the secondary mitigation has a balancing effect just for large applied current densities.

The second case occurs when the primary mitigation is directed to concentration overpotential and the secondary mitigation is directed to ohmic overpotential. When the primary mitigation is more effective for small applied current densities, the secondary mitigation acts in the concentration overpotential. On the other hand, when large current densities are applied, the opposite effect occurs

5.4 Optimal geometry

In practice, a VRFB operates at various current densities making it impossible to change the cell geometry for each of them. However, it is possible to determine an optimal geometry for the overall current densities by choosing the geometry that leads to the smaller variation of *VE* over the current densities. This was performed using the following definition:

$$\Delta VE = VE_{var}(L_{in}, L_{out}/L_{in}) \; - \; VE_{opt} \; * \qquad \textbf{(146)}$$

*Algorithm developed to calculate ΔVE is in Appendix A.

Where VE_{opt} is the constant optimum voltage efficiency for each applied current and $VE_{var}(L_{in},L_{out}/L_{in})$ is the voltage efficiency for each optimal geometry portrayed in **Figure 40**. For example, at 25 mA cm^{-2}, the VE_{opt} is equal to 88.83%, the $VE_{var}(7.57,0.57) = 88.83\%$, the $VE_{var}(5.25,0.89) = 88.40\%$ and the $VE_{var}(2.92,1.2) = 87.09\%$. Then, the ΔVE is equal to, respectively: 0, -0.43, -1.74%. **Figure 43**(a) shows the ΔVE for three values of applied current density versus the optimum geometric condition.

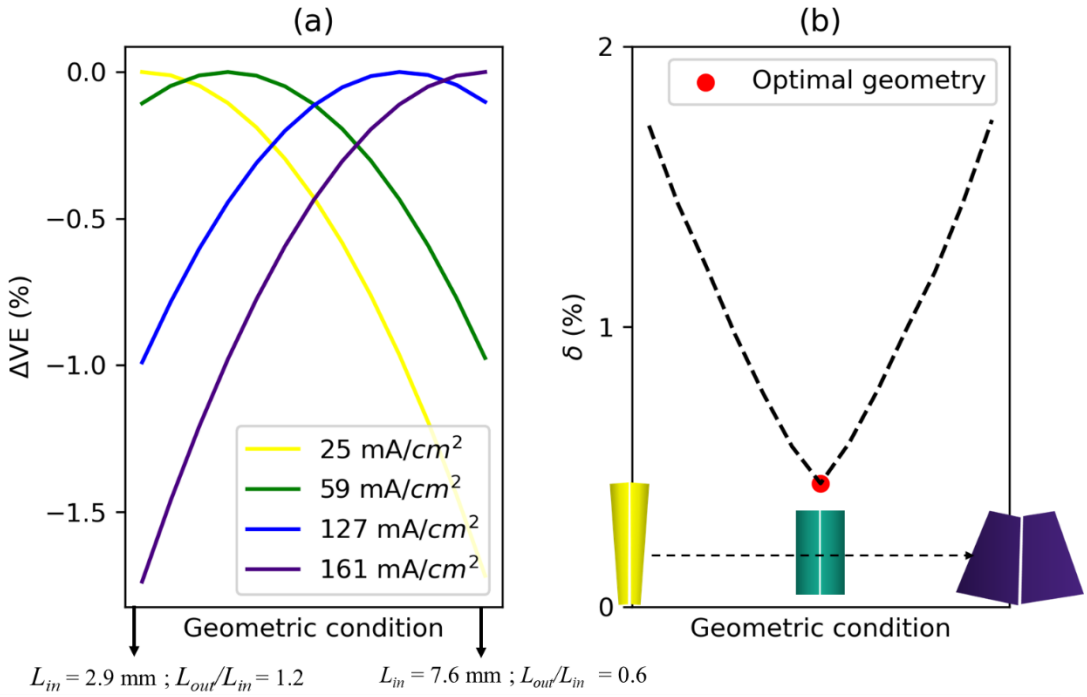


Figure 43. *(a) ΔVE values versus optimum geometric condition for different values of applied current density and (b) δ values versus optimum geometric condition*.*
**Algorithm developed to plot this graph is in Appendix A.*

The difference between the maximum and minimum value of ΔVE indicates the geometry that leads to the smaller variation of VE over the current densities. This parameter is defined as

$$\delta = max(\Delta VE) - min\ (\Delta VE)* \qquad \textbf{(147)}$$

**Algorithm developed to calculate δ is in Appendix A.*

Figure 43(b) shows a plot of δ versus the geometric condition. The minima correspond to the optimal geometry: L_{in} = 4.49 mm and L_{out}/L_{in} = 0.99. Thus, traditional geometries are better than trapezoidal geometries when the overall applied current densities are considered.

Thus, despite the previous works suggesting that trapezoid geometries – by employing $w_{out}/w_{in} < 1$ - leads to the better voltage efficiency [66,67], this is not observed for our study, which employs a trapezoid geometry varying L_{out}/L_{in}. Unlike w_{out}/w_{in} modification, which just modifies the concentration overpotential by accelerating the electrolyte inside the half-cell, the L_{out}/L_{in} controls the concentration overpotential by varying the active area per H and the ohmic overpotential by varying the electrode thickness. Indeed, the antagonistic consequences of both factors made the traditional

rectangular geometry more suitable to achieve better voltage efficiency (L_{out}/L_{in} = 0.99). Beyond that, an optimum electrode thickness can be defined by regulating these factors (L_{in} = 4.49 mm).

5.5 Validation of regression model

All implications showed in the above sections are based on the regression model obtained with 31 designed experiments. To confirm these implications, two validation approaches were carried out. The first validation (random validation) was performed with 1/3 of computational experiments and all variables were changing randomly. The second validation (validation of optimum responses) aims to confirm the values of maximum voltage efficiency for each applied current density. **Figure 44** shows the predicted versus simulated voltage efficiency of the training set and the validation sets.

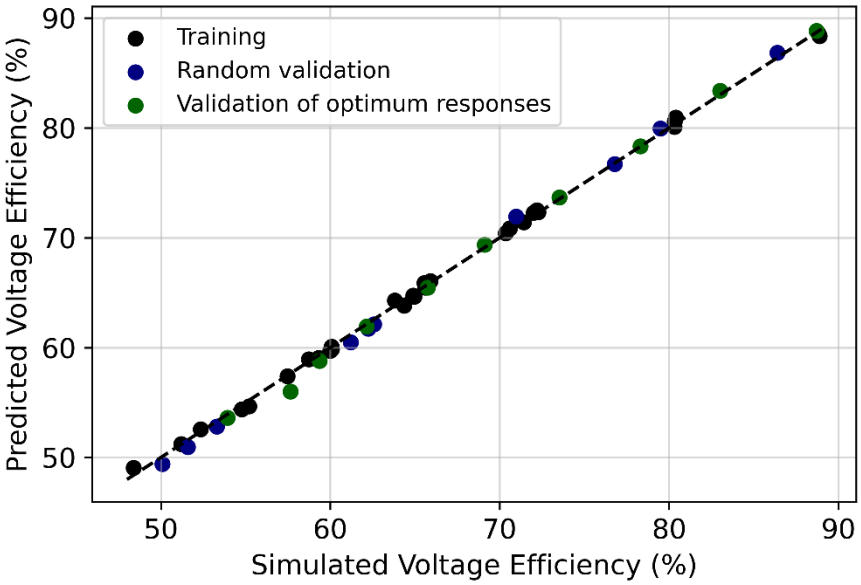


Figure 44. *Predicted versus simulated voltage efficiency for designed experiments and data validations*.*
**Algorithm developed to plot this graph is in Appendix A.*

The well-fitting of validation values were reached for voltage efficiency, ohmic and concentration overpotentials (see **Table 26**). The rooted mean squared error (RMSE) of voltage efficiency for both validations was large than those for the designed experiments set, but are still extremely low when compared with all variations in the experimental

space. Therefore, Validation 1 proved that the model is robust even in random choices of variables, especially for voltage efficiency and ohmic overpotential. And Validation 2 confirmed, with low error, the values of maximum voltage efficiency.

Table 26. *Figures of merit for validation**

	VE	η_o	η_c
R^2 *(random validation)*	*0.9976*	*0.9992*	*0.9856*
RMSE (random validation)	*0.58*	*1.59*	*4.05*
R^2 *(validation of optimum responses)*	*0.9971*	*0.9966*	*0.9970*
RMSE (validation of optimum responses)	*0.59*	*1.71*	*2.61*

*Algorithm developed to determine these metrics is in Appendix A

5.6 Characteristics of the study based on the method of approach

Table 27 shows the complete characteristics of Study 1 based on the method of approach.

Table 27. *Complete characteristics of Study 1 based on the method of approach.*

Step	*Description*	*Checklist*
(i)	*Experimental indication of a problem*	*Energy efficiency loss caused by overpotential and pumping energy consumption*
(ii)	*Development of a multi-physical model.*	*VRFB cell potential model (SoC cut-off)*
(iii)	*Validation of the multi-physical model.*	*Figure 20 (Successfully validated)*
(iv)	*Use of chemometric analysis.*	*Multivariate regression analysis*
(v)	*Validation of the chemometric analysis.*	*Figure 44 (Successfully validated)*
(vi) and (vii)	*Insights to develop a mitigation strategy*	*Section 5.3*

As we saw in Methodology, the problem of this study is the energy efficiency loss caused by overpotential (ohmic and concentration) and by pumping energy consumption. Indeed, we observed that the pumping energy consumption is low to decrease the battery efficiency in the studied scale. But, the ohmic and concentration overpotentials were very relevant to voltage loss and, consequent, energy efficiency loss in VRFBs.

Steps (ii) and (iii) are related to the choice of a suitable multi-physical model and its validation, respectively, as discussed in Methodology. And steps (iv) and (v) are related to the choice of a chemometric tool and its validation. In this case, the chemometric tool

was the multivariate regression analysis, which was validated by two types of approaches, as can be seen in **Figure 44.** The validation was considered successful because the predicted values from the regression model are very correlated to the values obtained from the multi-physical model.

The last steps ((vi) and (viii)), are related to the insights to develop a mitigation strategy. First, we identified the most important variables (L_{in}, $L_{out}/L_{in,}$ *and* i_{appl}), then we understood their pure and interaction effects on voltage efficiency. These insights led us to conclude that the optimum cell geometry depends on the applied current density. Thus, we proposed a method to calculate the overall optimal geometry, since it is not reasonable to modify the cell geometry for each operating condition. Beyond be applying in VRFBs, the proposed method can be further applied in any type and scale of RFB and for flow-design (flow-by or flow-through).

6 A robust mitigation strategy for capacity loss in VRFBs under different operating conditions

This chapter shows the results of Study 2. The first section shows a brief introduction to the study. The second section describes the first considerations about electrolyte viscosity and the limiting electrolyte. The third section shows the results of 2^3 factorial designs. The next section discusses the mitigation strategy based on the volume transfer between the electrolyte tanks. Section 6.5 brings the details about the validation of regression analysis. Finally, the last section describes the characteristics of Study 2, showing its main consequences.

6.1 Introduction to the chapter

Since the disbalancing of active species leads to the reversible capacity loss in VRFBs, we added robustness in an already experimentally proposed method of automatic rebalancing by volume transfer between the electrolyte tanks. First, we carried out a systematic study to investigate which variables control the capacity loss. Then, we built a regression model to predict the optimum value of flow velocity between the tanks. We innovated by (i) coupling multi-physical model and chemometric tools to investigate the capacity loss in VRFBs; (ii) predicting electrolyte viscosities from literature data and setting these values in the multi-physical models; (iii) identifying the most important operating conditions for capacity loss and explaining why they are important; (iii) proposing a method to calculate the optimum velocity flow between tanks to achieve the minimum capacity loss in VRFBs

Figure 45 shows a graphical abstract for Study 2.

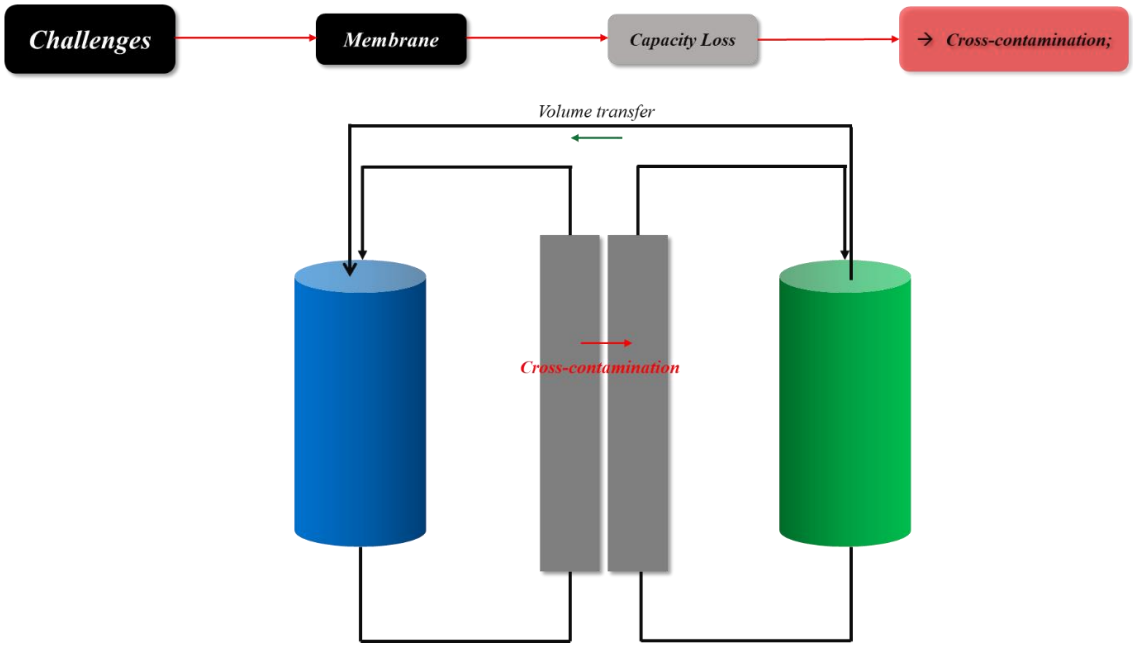


Figure 45. *Graphical abstract for Study 2.*

6.2 Preliminary considerations

6.2.1 Electrolyte viscosity

The concentration of active species and the concentration of supporting electrolytes influence the viscosity of electrolytes, as described by Song et al[72]. The electrolyte viscosity determines the pressure gradient between the half-cells and, as consequence, the cross-contamination by osmosis. Since the concentration of active species is a variable of this study, it is necessary to evaluate its effect on electrolyte viscosity.

The viscosity of each electrolyte was determined by a regression analysis using experimental data from literature: 69 points for the negative electrolyte[72,73] and 71 points for the positive electrolyte[72,73]. The experimental data were obtained for different values of SoC, the concentration of active species and, the concentration of supporting electrolyte. The values of electrolyte viscosity used in the multi-physical model were the mean value of electrolyte viscosity in the SoC window for each half-cell. The temperature was fixed at 298.15 K.

Figure 46 shows the predicted versus experimental viscosity for each electrolyte. The experimental data was well-fitted to the quadratic regression analysis for both positive and negative electrolytes.

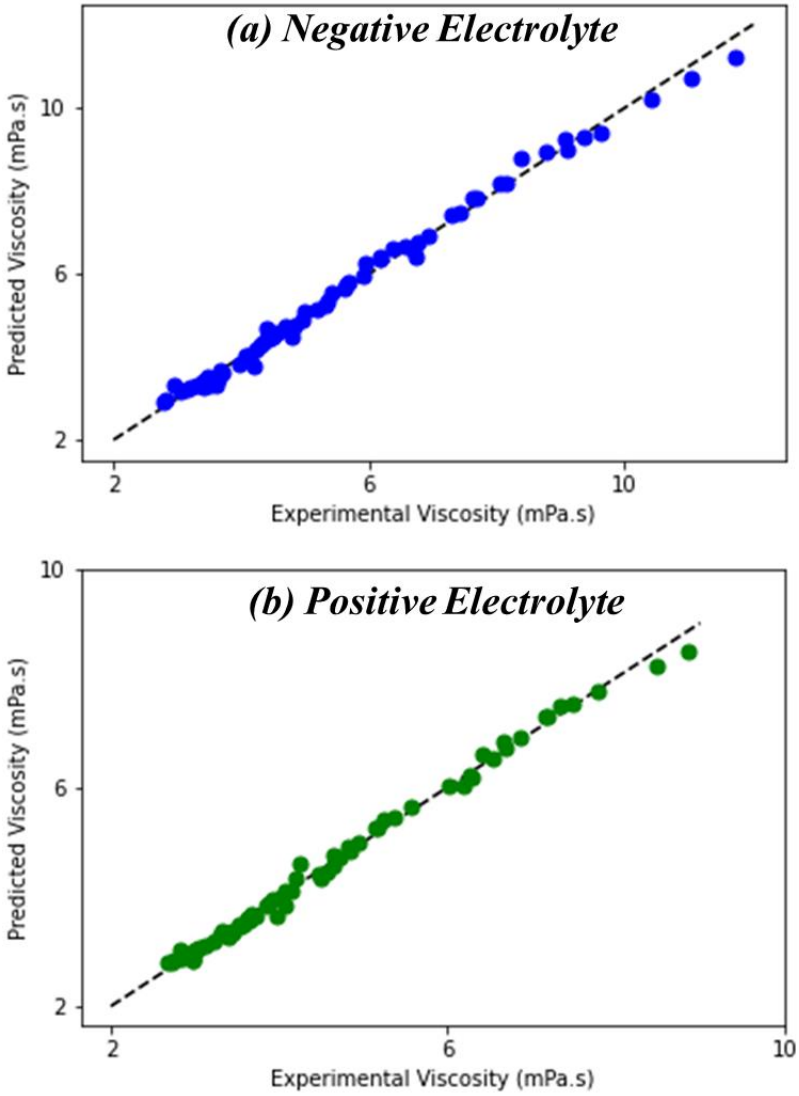


Figure 46. *Predicted versus experimental viscosities for (a) negative and (b) positive electrolytes*.*
**Algorithm developed to plot this graph is in Appendix B.*

The coefficients of determination (R^2) of the training set for both regressions are high, showing the strong correlation between predicted and experimental values. The high R^2 of cross-validation for both cases show the capacity and robustness of the quantitative model to predict the values of electrolyte viscosity.

6.2.2 The limiting electrolyte

In a VRFB, the cross-contamination leads to the imbalance of active species between the half-cells. The consequence of such imbalance across the cycles is visualized by the increasing of maximum negative SoC and decreasing of maximum positive SoC per cycle (see **Figure 47**).

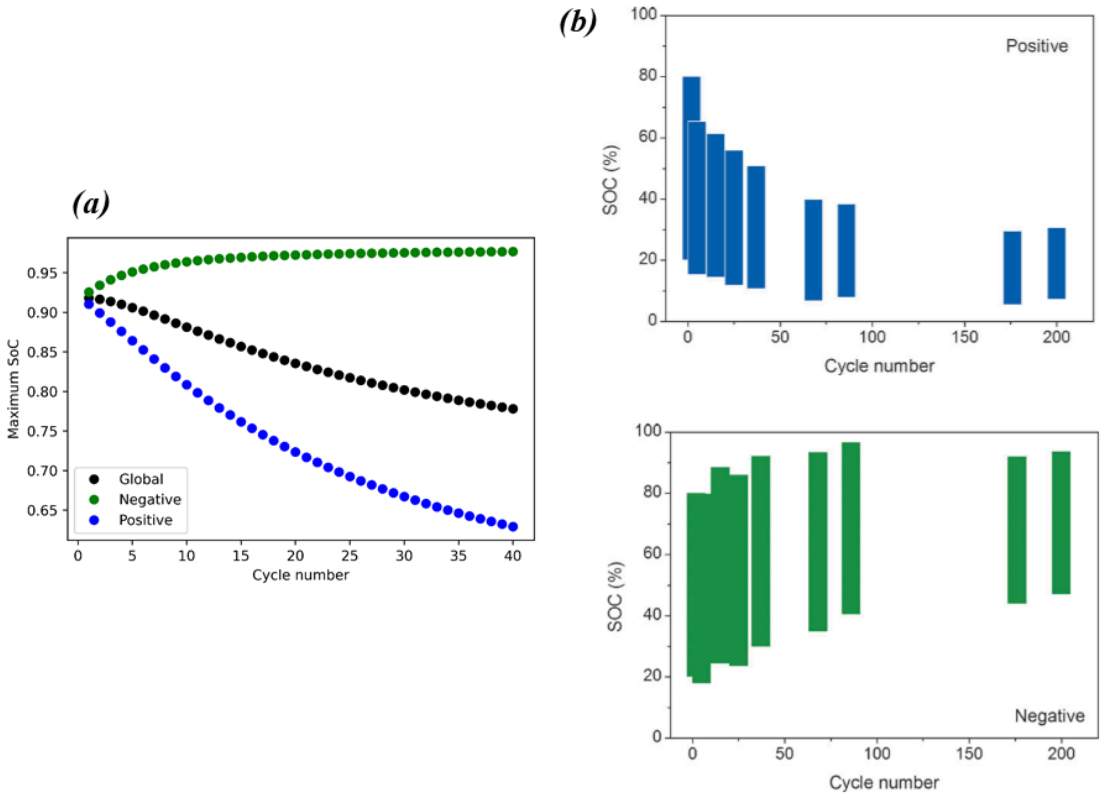


Figure 47. *Typical behavior of negative SoC, positive SoC, and global SoC along 40 charge-discharge cycles. (a) Results obtained from simulation* and (b) experimental results*[45].

The SoC behavior resulting from the simulation is in agreement with the experimental behavior portrayed by Luo et al.[45] (compare **Figure 47**(a) and **Figure 47**(b)).

Since the net cross-contamination occurs from the negative to the positive half-cell. Thus, the negative SoC decreases across the cycles due to: (i) the passage of species vanadium from one half-cell to another; and (ii) the higher diffusion coefficient of V^{3+} than V^{2+}:

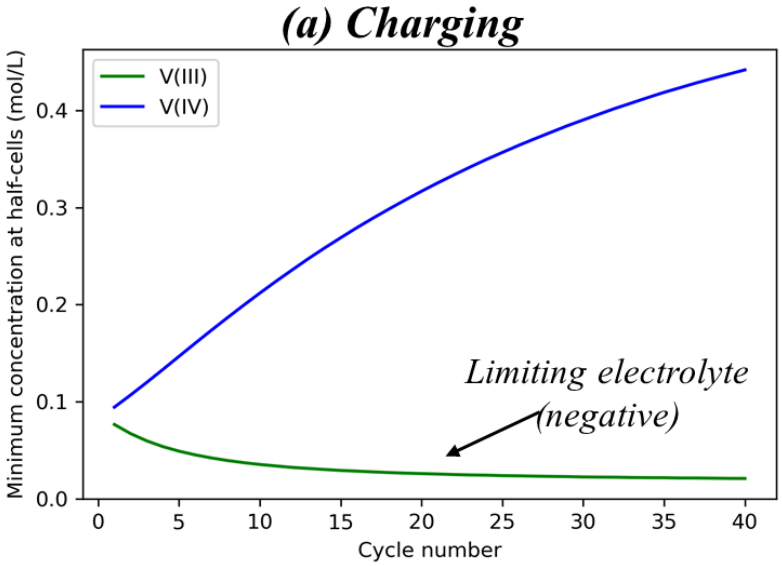
$$\uparrow SoC_{neg} = \frac{\downarrow c_{V(II)}}{\downarrow c_{V(II)} + \downarrow \boldsymbol{c_{V(III)}}} \qquad \textbf{(148)}$$

The excess of vanadium species in the positive half-cell is mostly composed of VO^{2+} (V(IV)) due to the self-discharge reactions. This explains why the positive SoC decreases across the cycles:

$$\downarrow SoC_{pos} = \frac{\downarrow c_{V(V)}}{\uparrow \boldsymbol{c_{V(IV)}} + \downarrow c_{V(V)}} \quad \textbf{(149)}$$

Where $c_{V(II)}$ and $c_{V(V)}$ are the dischargeable vanadium species and $c_{V(III)}$ and $c_{V(IV)}$are the chargeable vanadium species.

Thus, the negative electrolyte - lacking chargeable active species ($c_{V(III)} < c_{V(IV)}$) – limits the charging procedure of the battery (see **Figure 48**(a)). Whereas, the positive electrolyte – lacking dischargeable species *due to high self-discharge rate* ($c_{V(V)} < c_{V(II)}$– limits the discharging procedure of the battery (see **Figure 48**(b)).



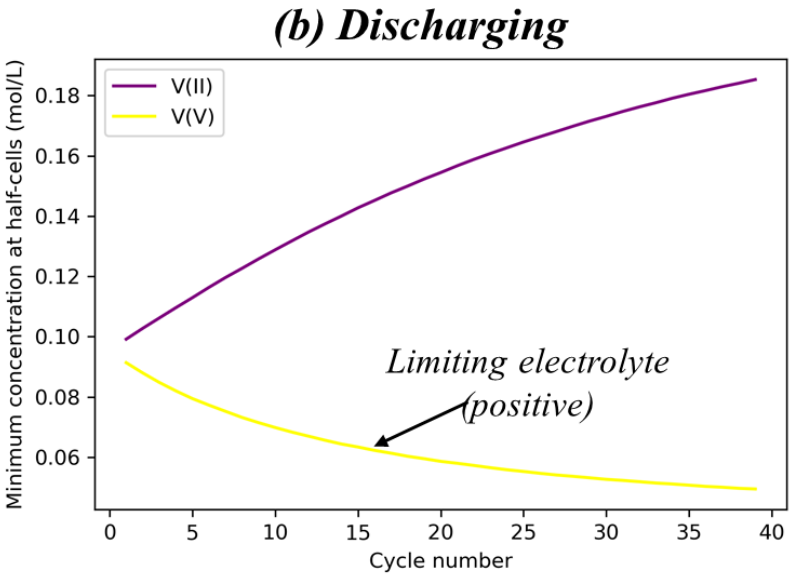


Figure 48. *Minimum concentration per cycle for charging (a) and discharging (b) procedures.*

Thus, the negative electrolyte is the limiting electrolyte in the charging procedure and the positive electrolyte is the limiting electrolyte in the discharging procedure.

6.3 Effects of variables on the capacity loss rate

Table 28 shows the values of electrolyte viscosity and capacity loss rate for the 2^3-factorial design (from designed experiments of **Table 16** in Methodology). The electrolyte viscosities are the predicted values set in the multi-physical model. The CL_{rate} is the main response of each designed experiment.

Table 28. *Responses of 2^3 factorial design (Study 2)*.*

ID	μ_{neg} *(mPa s)*	μ_{pos} *(mPa s)*	CL_{rate} *(%/cycle)*
1	*3.30*	*2.92*	***0.59***
2	*3.30*	*2.92*	***0.50***
3	*5.30*	*4.32*	***0.87***
4	*5.30*	*4.32*	***0.63***
5	*3.30*	*2.92*	***0.57***
6	*3.30*	*2.92*	***0.46***
7	*5.30*	*4.32*	***0.86***
8	*5.30*	*4.32*	***0.61***

*Algorithm developed to calculate these responses is in Appendix B.

Figure 49 shows the percentage of each effect on the response (capacity loss rate) and a probability graph of the effects.

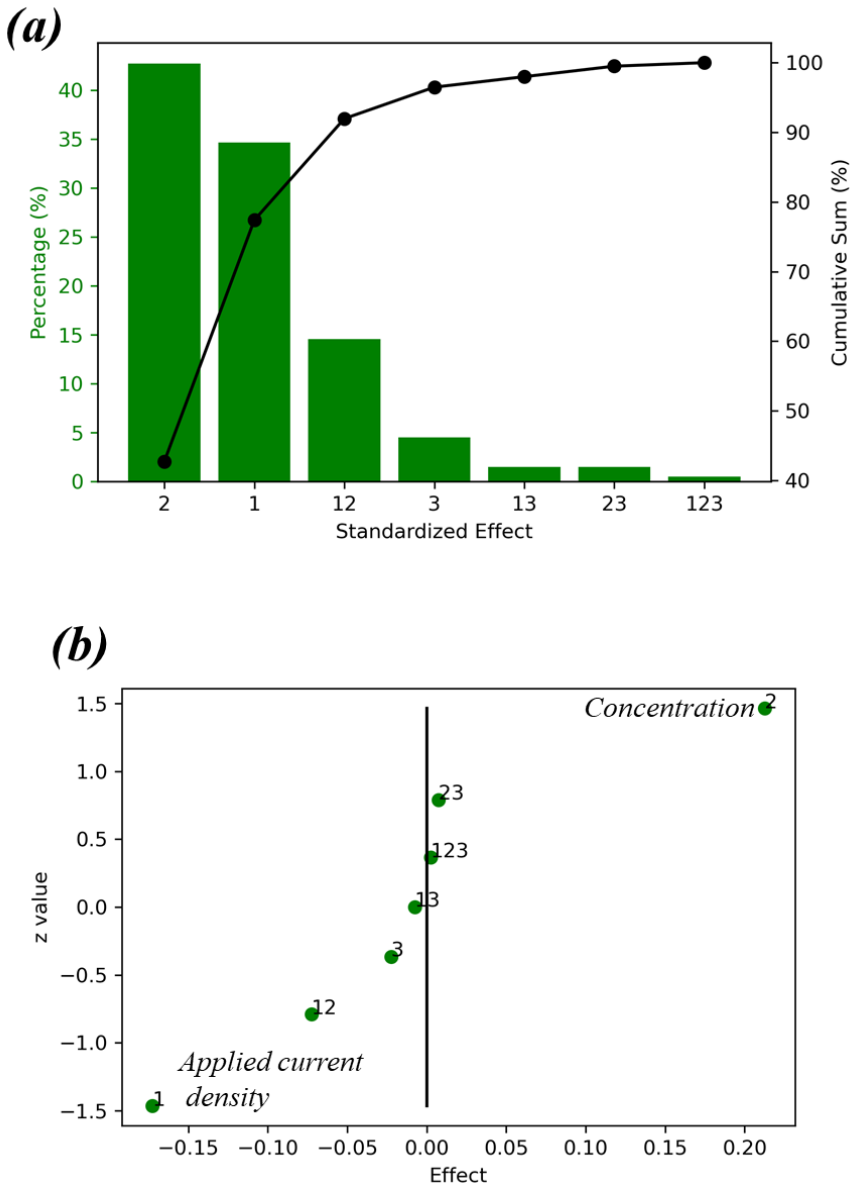
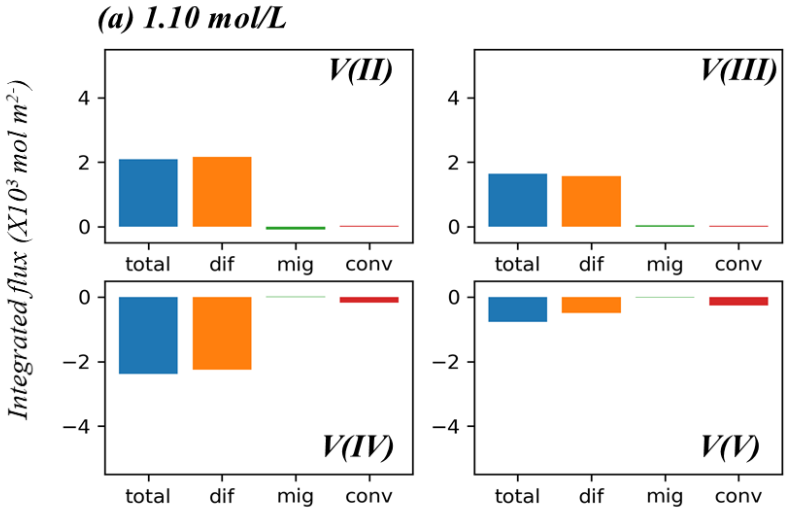


Figure 49. *(a) Percentage of each effect on the capacity loss rate and cumulative sum and (b) the probability graph of effects. (1) applied current density; (2) concentration of active species and; (3) volumetric flow. * *Algorithm developed to plot this graph is in Appendix B.*

The most relevant effects to the capacity loss rate were the concentration of active species (effect 2), the applied current density (effect 1), and the interaction effect (effect 12) between these two variables. The volumetric flow (effect 3) had less than 5% of contribution to the response, as well as the remaining secondary and tertiary effects.

Effect of concentration on capacity loss rate. The increase of capacity loss rate is about 0.20% per cycle when the concentration of active species goes from 1.1 to 1.8 mol L^{-1}. Thus, although the increased concentration of active species increases the theoretical capacity of the cell, it also leads to the acceleration of capacity loss. An explication for this effect is the increase of diffusional flux across the membrane due to a higher concentration gradient in the membrane/electrode interface when a larger concentration of active species is applied. **Figure 50** shows the integrated net fluxes - total, diffusional, migration, and convective - of vanadium species in the membrane/electrode interface for two different concentrations of active species: 1.10 and 1.80 mol/L.

The fluxes were integrated in the membrane|electrolyte interface region to obtain the quantity, in mols, of active species that cross the membrane (per unit of area) in each time of the simulation. To obtain the net fluxes, we just took the mean values of fluxes in all the calculated times, considering the change of direction in migration and convection fluxes.



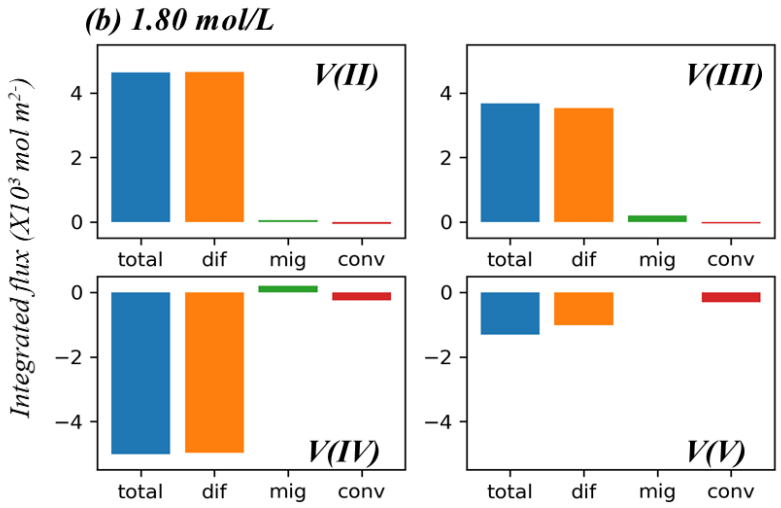


Figure 50. *Integrated net fluxes (total, diffusional, migration, and convective) across membrane/electrolyte interface for vanadium species with the initial concentration of (a) 1.10 and (b) 1.80 mol L^{-1}. dif: diffusional flux; mig: migration flux; conv: convective flux and; total: the sum of the diffusional, migration and convective flux*.*
**Algorithm developed to plot these graphs is in Appendix B.*

It is possible to observe the increase of diffusional flux for the four vanadium species when the concentration of active species goes from 1.1 to 1.8 mol L^{-1}. On the other hand, the net migration and convective fluxes vary slightly for these cases. The net convective and migration fluxes are close to 0 because they change direction depending on the charging or discharging procedure.

The higher diffusional flux for 1.8 mol L^{-1} accelerates the variation of the V(V) limiting concentration during discharge, leading to higher capacity loss, as seen in the dotted lines of **Figure 51.**

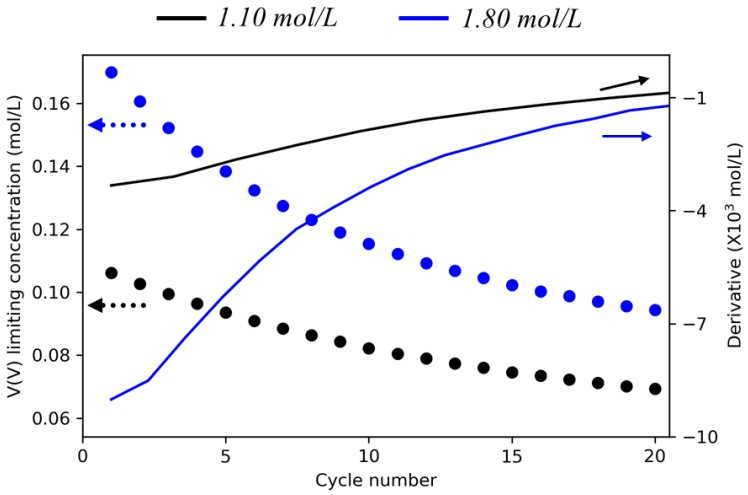


Figure 51. *The V(V) limiting concentration in discharging procedure for 1.10 and 1.80 mol L^{-1} initial concentration of active species*.*
**Algorithm developed to plot these graphs is in Appendix B.*

This acceleration in the variation of V(V) limiting concentration is highlighted by its derivative in the function of cycle number (solid lines in the above figure).

Thus, the increase of concentration of active species increases the diffusional fluxes across the membrane and accelerates the consumption of the limiting active species. As a consequence, the capacity loss across the cycles increases as concentration increases. **Figure 52** schematizes the effect of the concentration of active species on capacity loss.

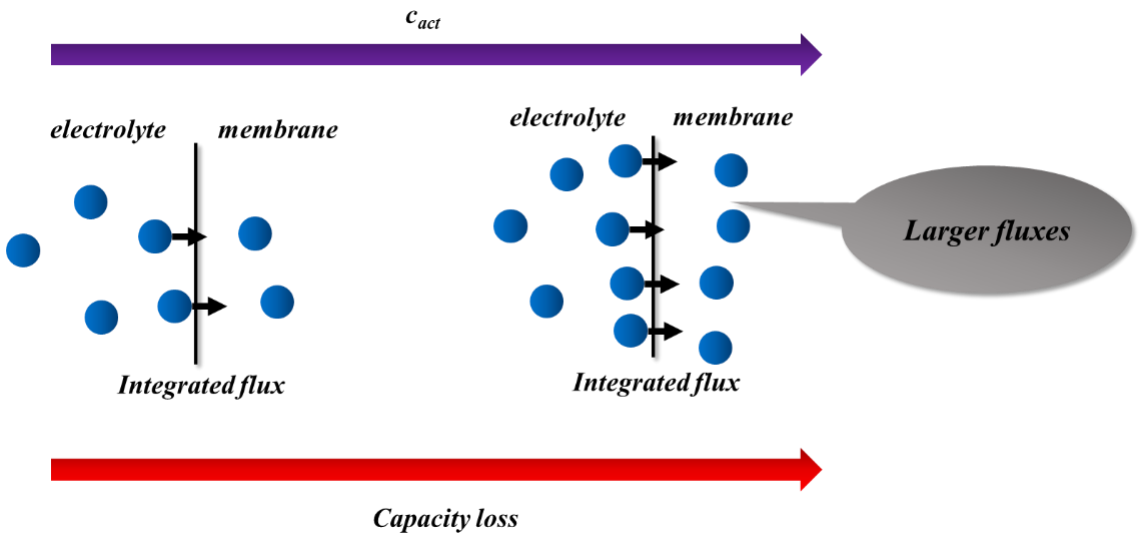


Figure 52. *Schematic representation of effect of concentration of active species on the capacity loss due to the increase of fluxes across membrane.*

Effect of applied current density on capacity loss rate. The decrease of capacity loss rate is about 0.15% when the applied current density goes from 50 to 100 mA cm^{2-}. Despite larger currents increasing the concentration and potential gradients in the membrane/electrolyte interface, these cases also correspond to shorter charge/discharge cycles. Thus, the integrated flux over time across the interface is smaller when larger applied currents are applied (see **Figure 53**).

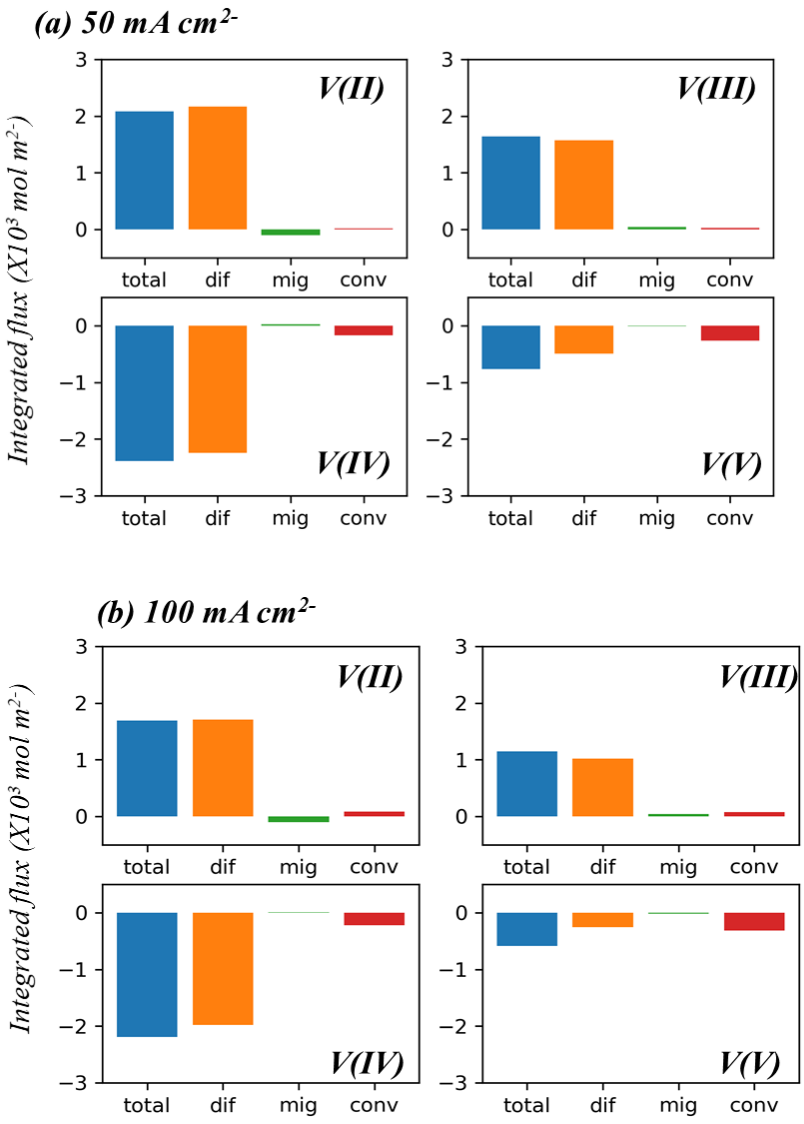


Figure 53. *Integrated net fluxes (total, diffusional, migration, and convective) across membrane/electrolyte interface for vanadium species with an applied current density of (a) 50 and (b) 100 mA cm^{2-}. dif: diffusional flux; mig: migration flux; conv: convective flux and; total: the sum of the diffusional, migration and convective flux*.*
**Algorithm developed to plot these graphs is in Appendix B.*

The different integrated flux leads to different limiting concentration profiles, as shown in **Figure 54.** As smaller currents lead to smaller integrated fluxes, the decrease of limiting concentration per cycle is smaller too, explaining why the capacity loss rate decreased.

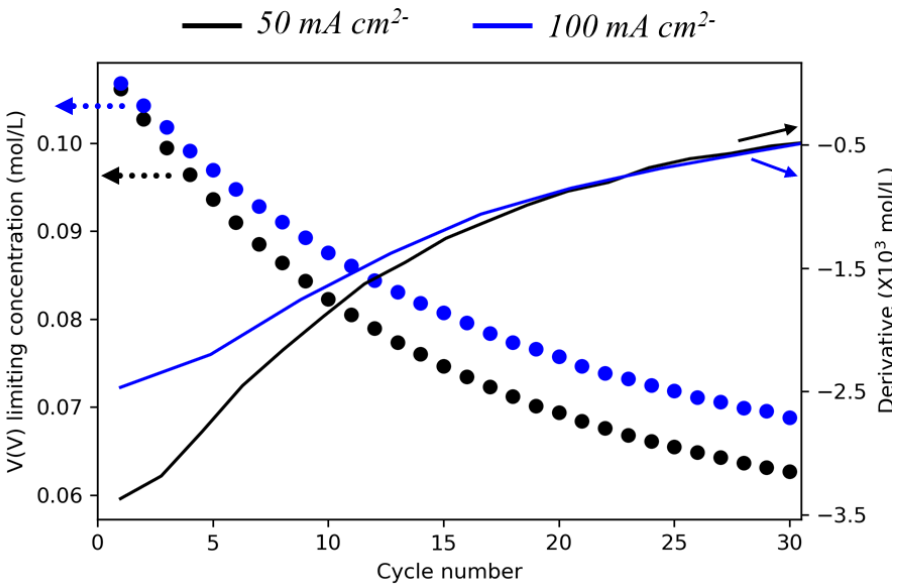


***Figure 54**. The V(V) limiting concentration in discharging procedure for 50 and 100 mA cm^{2-}. * *Algorithm developed to plot these graphs is in Appendix B.*

Thus, the increase of applied current density shortens the charge/discharge cycles, making fewer species cross the membrane and decreasing the capacity loss per cycle. As consequence, the capacity loss across the cycles decreases. **Figure 55** schematizes the effect of the concentration of active species on capacity loss.

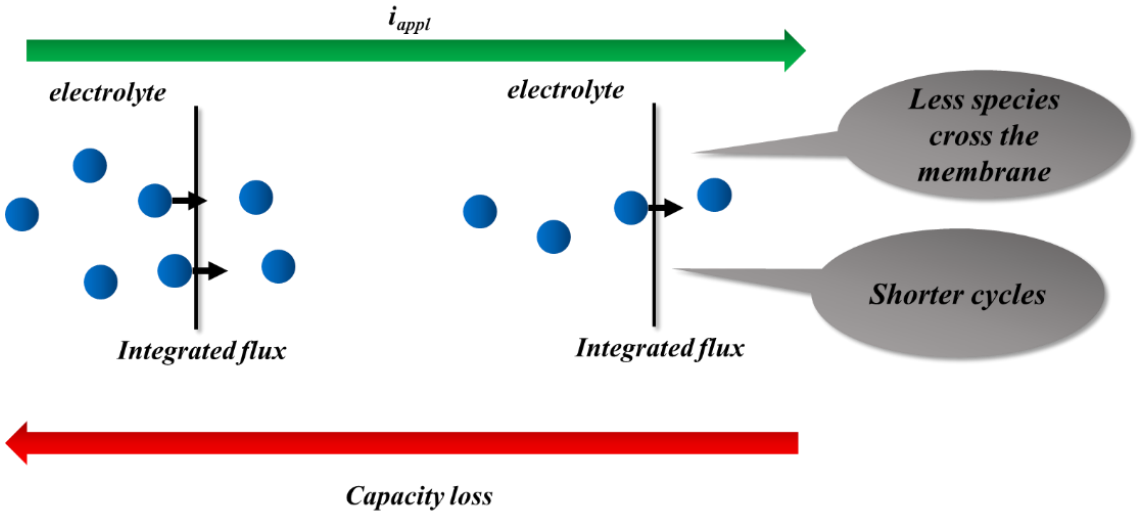


***Figure 55.** Schematic representation of the effect of applied current density on the capacity loss due to the short of charge/discharge cycles.*

Interaction effect between concentration and current on capacity loss rate. The interaction effect 12 is about -0.05% of capacity loss per cycle. Thus, when both variables

– applied current density and concentration of active species – are at higher levels, there is an additional decrease of 0.05% in the capacity loss rate.

Since the applied current density decreases the integrated net fluxes and the concentration of active species increases the integrated net fluxes; this interaction effect indicates that the increasing of applied current densities is sufficient to decrease the integrated net fluxes caused by the increase of concentration. Thus, when a high concentration of active species is used, the use of larger applied current densities is an option to mitigate the increased capacity loss rate. **Figure 56** schematizes the interaction effect between applied current density and concentration of active species on capacity loss.

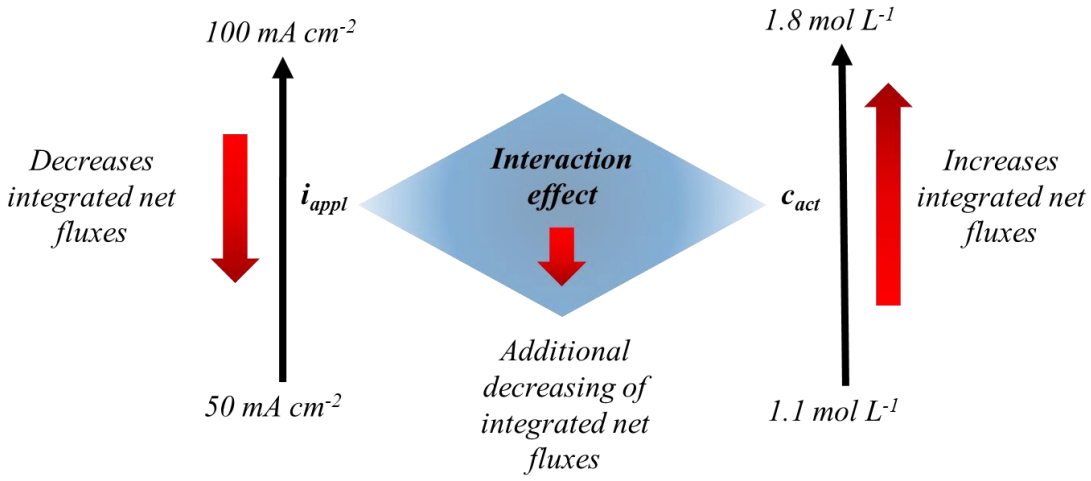


Figure 56. *Schematic representation of interaction effect between applied current density and concentration of active species.*

6.4 Mitigation strategy

6.4.1 Evaluating the volume transfer between electrolyte tanks

Since the variables that affect the capacity loss are known (concentration of active species and applied current density), is possible to apply the mitigation strategy. This strategy consists of the volume transfer between the electrolyte tanks in the inverse direction of net cross-contamination (see **Figure 57**).

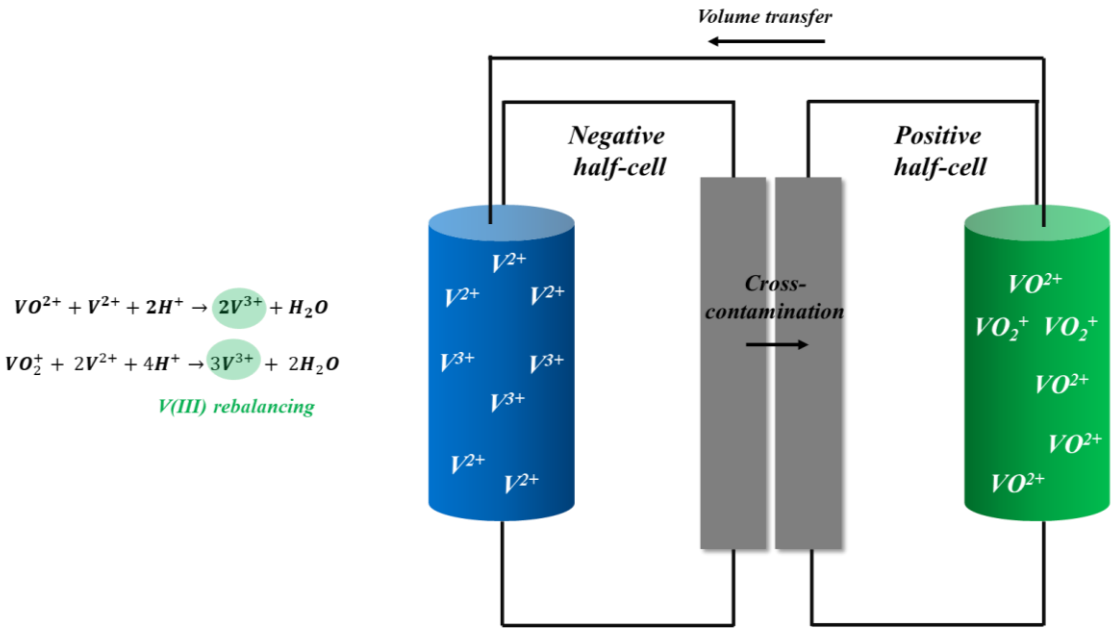


Figure 57. *Scheme of mitigation strategy of capacity loss in VRFBs based on volume transfer between electrolyte tanks.*

This strategy is only possible because of self-discharge reactions that take place in the negative half-cell. The VO_2^+ (V(V)) and VO^{2+} (V(IV)) react with V^{2+} - in excess in the negative half-cell - producing V^{3+}, initially in a small quantity. This process leads to the rebalancing of V^{3+} in the negative half-cell especially by the consumption of V(IV). Because of this, it is expected partial or total mitigation of capacity loss by using this strategy. However, the suitable velocity of volume transfer between the electrolyte tanks must be determined. For this, we changed the parameter v_p in one order of magnitude to evaluate the capacity loss (see **Figure 58**).

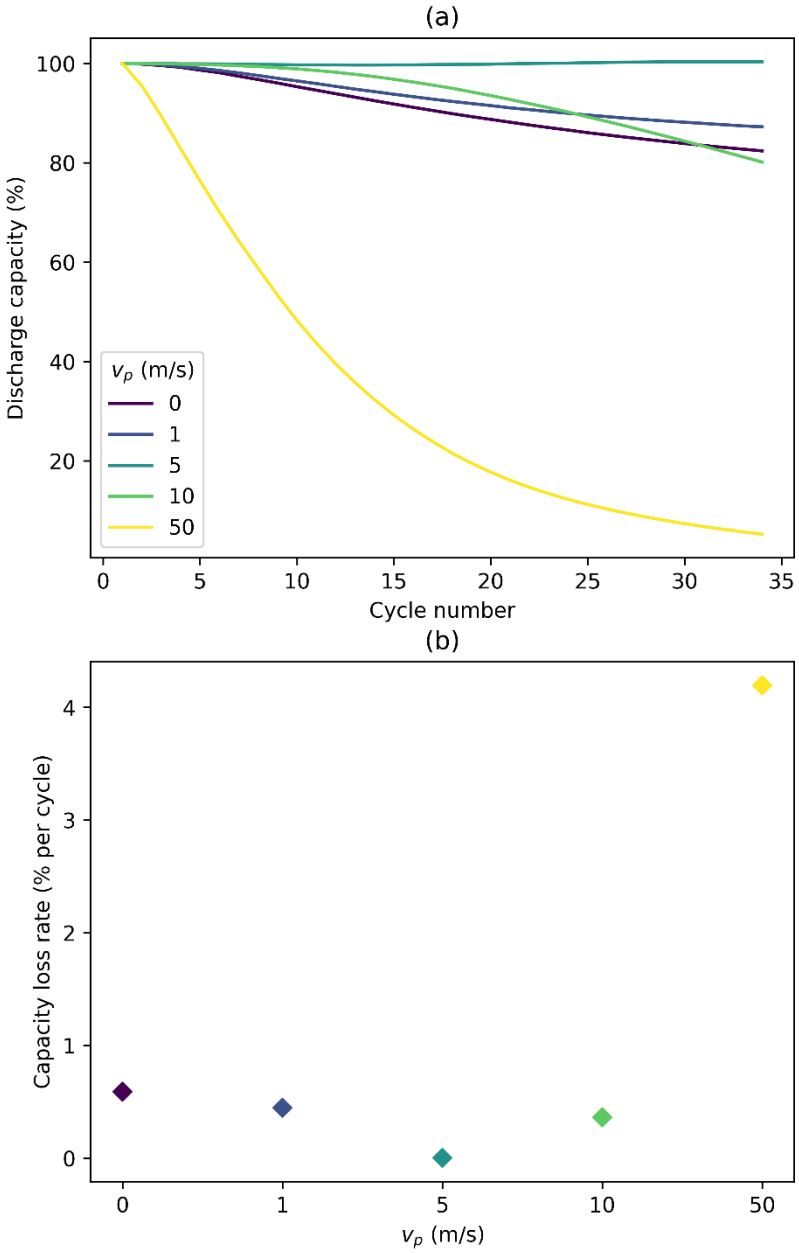


Figure 58. *(a) Discharge capacity versus cycle number for different values of velocity between tanks and (b) capacity loss rate versus v_p parameter.**
**Algorithm developed to plot these graphs is in Appendix B.*

For the case without volume transfer ($v_p = 0$) it was observed the expected capacity loss at a rate of 0.59% per cycle. For the cases with volume transfer between tanks, the capacity loss rate decreased slightly when $v_p = 1$ *m/s* and approximates to 0% when $v_p = 5$ *m/s*. This indicates that capacity loss can be completely mitigated by the proposed

strategy. But it is necessary to choose properly the velocity between tanks because if high velocities are used, the capacity loss rate increases again (see the cases for $v_p = 10$ and *10 m/s*). Thus, there is a region of the minimum that indicates the optimum velocity between tanks.

6.4.2 Searching optimum flow velocity between tanks

As we saw in the previous sections, the concentration of active species and the applied current density affect the capacity loss in a VRFB cell. We also observed that is a region of optimum velocity between tanks for a given operating situation. The purpose now is to turn our strategy general by finding the optimum velocity between tanks for any condition of concentration and current. For this, a regression analysis was performed based on a Doehlert design of experiments (from **Table 17** in Methodology). **Table 29** shows these results.

Table 29. *All responses from Doehlert design (Study 2)**

ID	*CL_{rate} (%/cycle)*[a]	*CL (%)*[b]
1	*0.0010*	*0.04*
2	*0.4515*	*19.42*
3	*0.0767*	*4.60*
4	*0.3621*	*16.30*
5	*0.0924*	*2.13*
6	*0.3987*	*9.17*
7	*0.1130*	*7.23*
8	*0.1532*	*5.82*
9	*0.1024*	*4.81*
10	*0.1676*	*7.71*
11	*0.0370*	*2.63*
12	*0.1382*	*5.53*
13	*0.0109*	*0.25*
14	*0.0052*	*0.18*
15	*0.0589*	*1.41*

[a]Calculated based on the first 20th cycles
[b]|100-discharge capacity| at 20th cycle
*Algorithm developed to calculate responses is in Appendix A

Figure 59 shows three stacked contour plots for capacity loss of v_p *versus* applied current density for three different concentrations of active species.

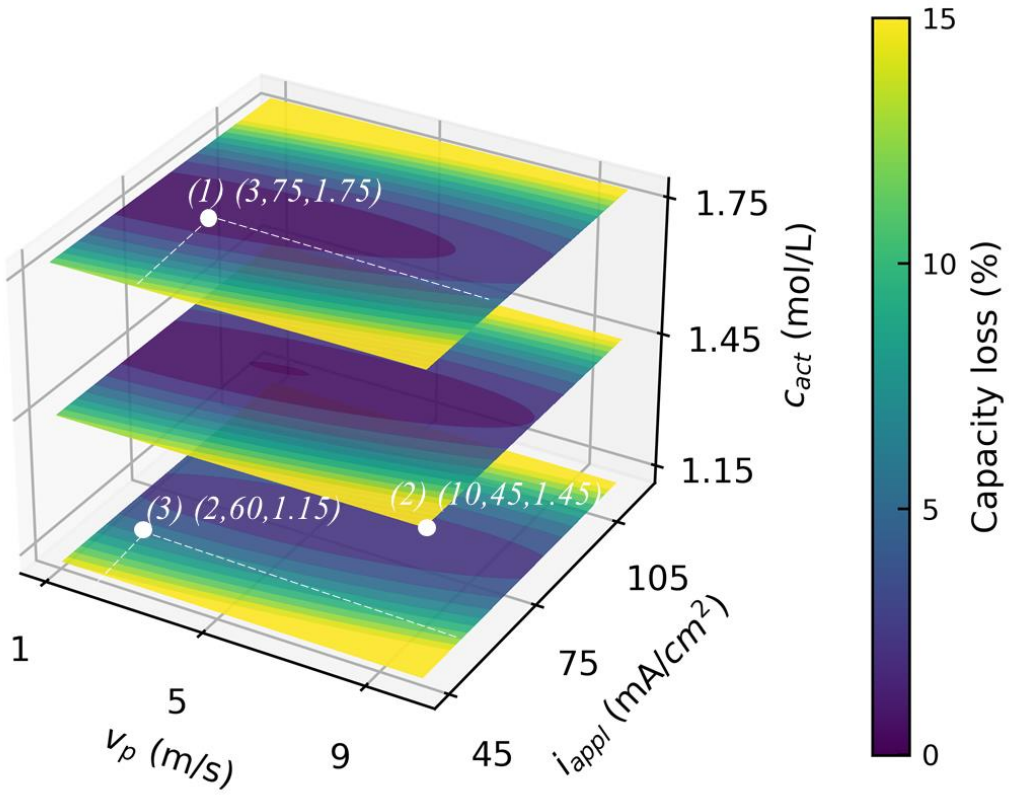


Figure 59. *Contour graphs stack (4D plot) of v_p versus applied current density versus concentration of active species for capacity loss. The capacity loss was determined by |100 – discharge capacity at 20th|*.*
**Algorithm developed to plot these graphs is in Appendix B.*

Three distinct cases can be identified in the above figure. The coordinates are described as (v_p(m/s), i_{appl} (mA/cm^2), c_{act}(mol/L)).

(1) (3, 75, 1.75). The chosen velocity between tanks leads to a capacity loss equal to zero, in a global minimum region. If a higher value of v_p is used (x-direction), the capacity loss increases.

(2) (10, 45, 1.45). The chosen velocity between tanks leads to a larger capacity loss located in the maximum region in the x-direction. If a low value of v_p is used (x-direction), the capacity loss decreases but does not reach zero.

(3) (2, 60, 1.15). The chosen velocity between tanks leads to a small capacity loss different from zero, in the minimum region in the x-direction. If a high value of v_p is used (x-direction), the capacity loss increases.

Thus, not all combinations of i_{appl}/c_{act} will lead to null capacity loss, but regions of the minimum are always identified in this experimental space. That way, it is possible to identify the optimum velocity between tanks for any combination of variables by simply

choosing the value of applied current density and concentration of active species and identifying the region of minimum capacity loss. **Figure 60** shows this process for the three cases under study.

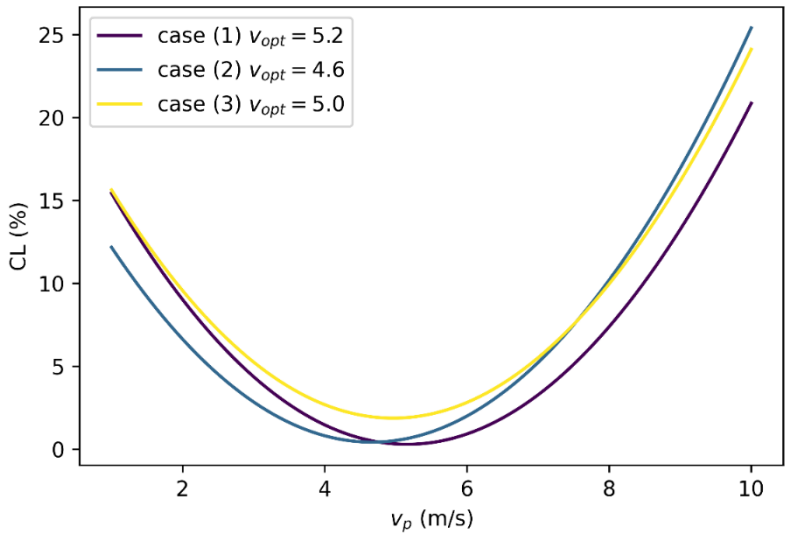


Figure 60. *Capacity loss versus v_p for the three different cases. **
**Algorithm developed to plot these graphs is in Appendix B.*

The optimum velocity between tanks varies slightly for the three cases portrayed in the above figure. Thus, despite applied current density and concentration of active species influencing the capacity loss, the optimum velocity between tanks is very close to a range of variable values.

As previously mentioned, just for case (1) and case (2) the optimum velocity between tanks led to null – or close to zero – capacity loss. In case (3), there was a minimum in capacity loss, but it is different from zero (close to 2%).

6.5 Regression analysis validation

The validation of regression analysis was performed using a random combination of variables. **Figure 61** shows the predicted versus simulated capacity loss for the test and train sets. The train set is used to determine the coefficients of the regression model and the test set is used for validation.

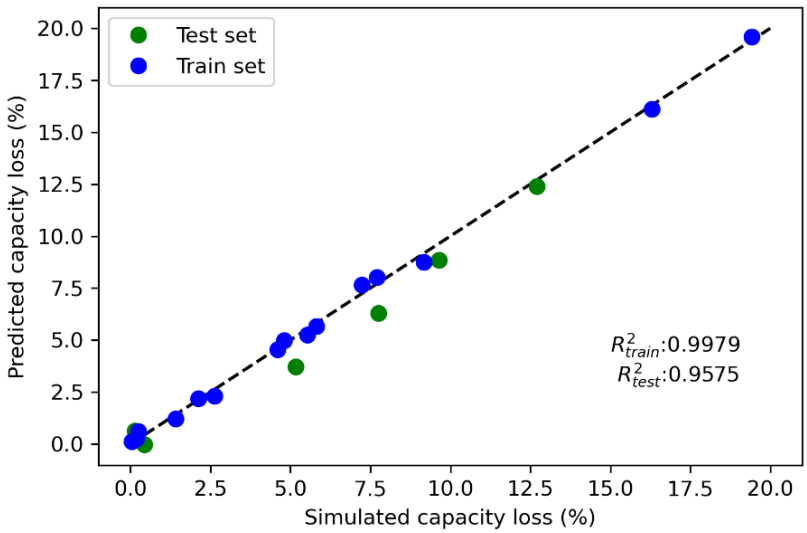


Figure 61. *Predicted versus simulated capacity loss for test and train sets*.*
**Algorithm developed to plot this graph is in Appendix B.*

The coefficient of determination (R^2) for the train set shows a well-fitting between simulated and predicted capacity loss. The R^2 for the test set demonstrated the capacity of the regression analysis in predicting the capacity loss even in random changes in the studied variables.

6.6 Characteristics of the study based on the method of approach

Table 30 shows the complete characteristics of Study 2 based on the method of approach.

Table 30. *Complete characteristics of Study 2 based on the method of approach.*

Step	*Description*	*Checklist*
(i)	*Experimental indication of a problem*	*Capacity loss caused by cross-contamination*
(ii)	*Development of a multi-physical model.*	*VRFB capacity loss model (Voltage cut-off)*
(iii)	*Validation of the multi-physical model.*	*Figure 19 (Successfully validated)*
(iv)	*Use of chemometric analysis.*	*Factorial design and regression analysis*
(v)	*Validation of the chemometric analysis.*	*Figure 61 (Successfully validated)*
(vi) and (vii)	*Insights to develop a mitigation strategy*	*Section 6.3*

As we saw in the Methodology, the problem of this study is the capacity loss caused by the cross-contamination at the membrane interface. A suitable multi-physical model to simulate this condition was successfully developed and validated. For this study, two chemometric analysis was carried out:

(i) Factorial design, for the determination of the most important variables that affect the capacity loss in VRFBs. These variables are the applied current density and the concentration of active species.

(ii) Regression analysis, for the testing the robustness of the proposed mitigation strategy by predicting the values of capacity loss caused by the applied current density, concentration of active species, and the flow velocity between the electrolyte tanks. The validation of regression analysis was made by comparing the simulated and predicted capacity losses. This validation was considered successful because the predicted values from the regression model are very correlated to the values obtained from the multi-physical model.

The proposed method allowed us to understand the impacts of operating conditions on VRFBs' capacity loss and how the mitigation strategy behaved in these conditions. (steps (vi) and (vii)). Thus, we contributed to an already proposed mitigation strategy[78] by studying it in several conditions and by providing a theoretical background to show the potentialities of this strategy. By this process, we also provide a method to calculate the optimum conditions to achieve the smallest possible capacity loss and, as consequence, improve the operational life of VRFBs.

7. Insights into the effects of active species properties on the voltage efficiency of redox flow batteries

This chapter shows the results of Study 3. The first section shows an introduction to the study. The second section describes how real and different active species behave in the RFB general model. The last section brings the results of 2^{5-1} fractional factorial design, discussing the consequences of choosing different active species for RFBs and showing which properties are more important.

7.1 Introduction to the chapter

Despite the use of metal-based active species (e.g., vanadium), the use of organic active species – or redox organic molecules (ROMs) – for RFB application is highly targeted due to the high adaptability of chemical and physical properties[106,107]. This allows the modification of properties aiming at the improvement of performance. In this section, we discuss the main characteristics in the choice and development of active species and show their consequences on RFBs' performance.

(i) Solubility. Solubility is a factor of great interest in RFBs, as it determines the magnitude of the capacity and energy density of the system defined as, respectively:

$$C = nFc_{act} \quad \textbf{(150)}$$

$$E = UC \quad \textbf{(151)}$$

Where *n* is the electrons transferred in the electrochemical reaction, *F* is the Faraday's constant, c_{act} is the concentration of active species and *U* is the theoretical cell redox potential. The concentration of active species is related to solubility since the solubility dictates the maximum concentration.

In the context of organic compounds, given their functional adaptability, solubility can be improved so that there is an increase in solute-solvent interactions[29] There are still other factors that govern the solubility of active species, such as the temperature of the system, its state of charge (oxidized or reduced), and competition with the supporting electrolyte used.

(ii) Redox Potential. As well as solubility, the redox potential also determines the magnitude of the energy density **(Equation 151**) and, additionally, the power density:

$$P = \frac{i_d U}{a} \qquad \textbf{\textit{(152)}}$$

Where i_d is the discharging current and *a* is the active area of the battery.

To improve both metrics is necessary to choose anolytes with high negative redox potential ($E_{anolyte} \ll 0$) and catholytes with positive redox potential ($E_{catholyte} \gg 0$) since,

$$U = E_{catholyte} - E_{anolyte} \qquad \textbf{\textit{(153)}}$$

For ROMs, the redox potential is governed by the energy levels of the HOMO (oxidation) and LUMO (reduction) orbitals[8]. Once again, the functional adaptation allows for changes in the energies of these boundary orbitals, allowing the improvement of the redox potential. The principle of this modification is based on functionalization using electron-donating (EDG) or electron-withdrawing groups (EWG). In general, EDG provides electrons to the redox centers resulting in a decrease in potential[107], whereas EWG shift the redox potential to more positive values due to an increase in the electron affinity of the molecule [107]. **Figure 62** shows an example of this type of modification for pyrylium ions with different substitutions in R_2. The data was obtained from literature[108].

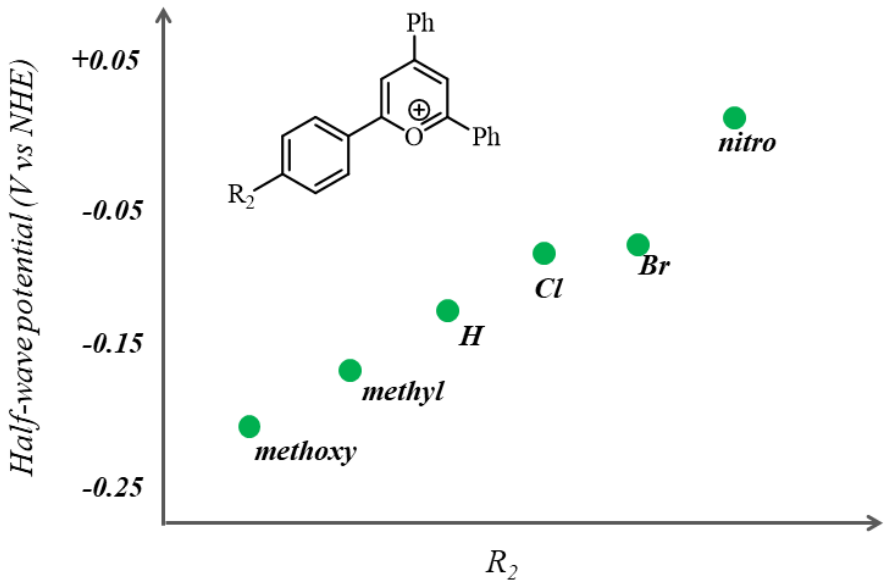


Figure 62. *Relationship between the half-wave potential in cyclic voltammetry and the substituent R_2. Data from*[108].

The nitro R_2-substitution increases the redox potential in 140 mV regarding H R_2-substitution, whereas the methoxy substitution in R_2 decreases the redox potential in 80 mV. Thus, the methoxy-substituted pyrylium ion is a better anolyte than the H-substituted. And the nitro-substituted pyrylium ion is a batter catholyte than the H-substituted.

(iii) Stability. The chemical stability of the products generated in the electron transfer process (oxidation or reduction) is one of the biggest challenges when it comes to the application of ROMs in RFBs. In many cases, the loss or gain of electrons by the molecule causes the formation of unstable radicals, which react generating electrochemically irreversible species[30,31]. Such chemical processes will be reflected in the loss capacity across the cycles.

These three properties, solubility, redox potential, and stability of electron-generated radicals are the target properties in the choice of a suitable active species for RFB application, due to the metrics discussed above. However, there is a lacking of understanding in:

(i) How solubility and redox potential affects the voltage efficiency in RFBs;

(ii) How mass (diffusion coefficient) and electron transfer (standard electrochemical constant rate) properties also affect the RFB's performance;

(iii) Which are the most relevant target properties in the choice of suitable active species for RFB application.

Based on the aforementioned information, we carried out a study to investigate how active species properties affect the voltage efficiency and, as a consequence, the energy efficiency of RFBs. We innovated by (i) coupling multi-physical model and chemometric tools to investigate the effect of active species properties in RFBS; (ii) developing a multi-physical model capable of simulating any combination of anolyte/catholyte to obtain performance data; and (iii) providing a robust theoretical background to understand the impacts of active species choosing in RFBs' performance.

Figure 63 shows a graphical abstract for Study 3.

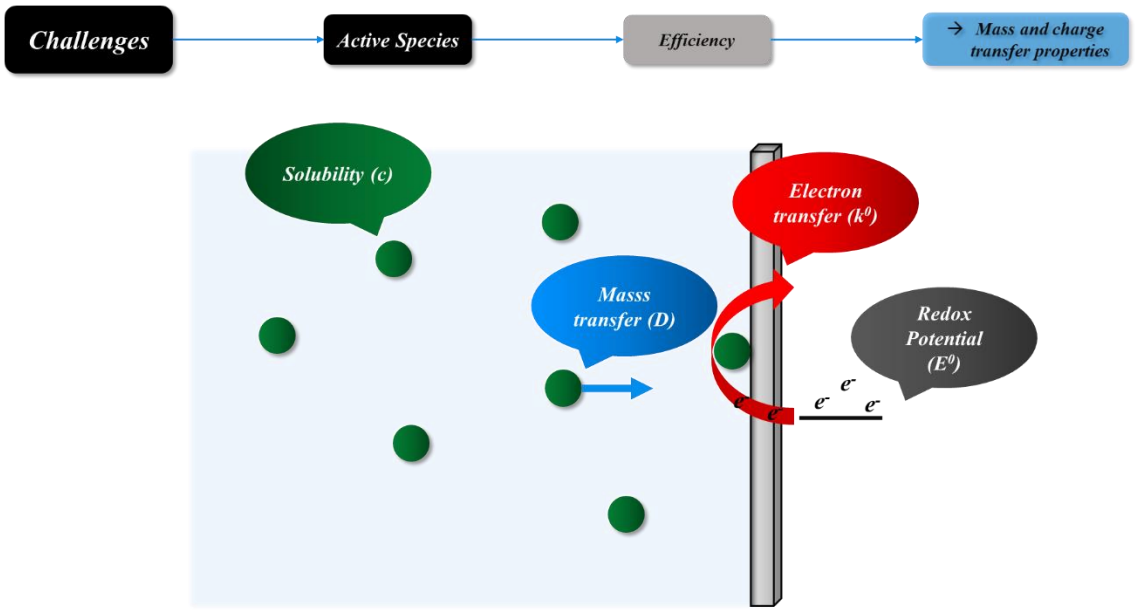


Figure 63. *Graphical abstract for Study 3.*

7.2 Analyzing real active species in the set boundary conditions

The starting point to understand the effects of active species properties on voltage efficiency is to simulate how real active species behave in the set boundary conditions.

Figure 64 shows the voltage efficiency versus applied current density for the three studied systems.

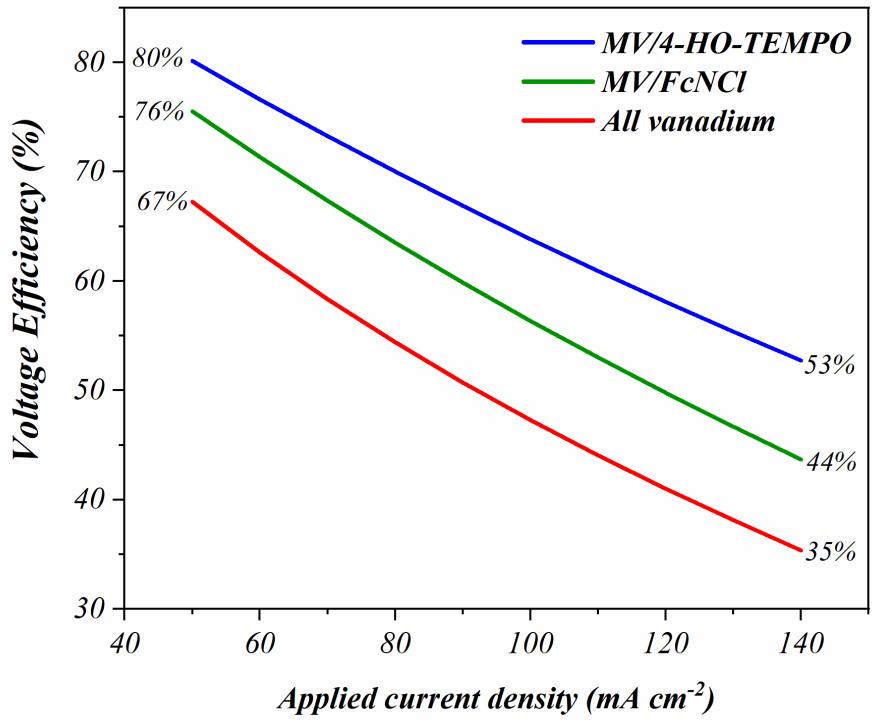


Figure 64. *Voltage efficiency versus applied current density for three combinations of anolytes and catholytes simulated in a steady-state multi-physical model.*

Considering the same anolyte MV (methyl-viologen), the change of catholyte from 4-OH-TEMPO to FcNCl modified the behavior of voltage efficiency versus applied current density. For MV/4-OH-TEMPO the voltage efficiency goes from 80% at 50 mA cm^{2-} to 53% at 150 mA cm^{2-}, a difference of 27%. Whereas in the MV/FcNCl, the voltage efficiency goes from 76% to 44% in the same conditions, a difference of 32%. 4-OH-TEMPO has a larger diffusion coefficient, standard electrochemical rate constant, and standard reduction potential (see **Table 31**). This explains why the combination MV/4-OH-TEMPO presented an improved behavior compared to MV/FcNCl, but does not identify which active species property has the important role in performance. To answer this last question, it is necessary a systematic study, as shown in the next section.

The all-vanadium system shows an even lower voltage efficiency, especially because of the sluggish electrochemical kinetics of both redox couple V^{2+}/V^{3+} and VO^{2+}/VO_2^{+}. There is a big difference between the all-vanadium and MV/FcNCl voltage efficiency considering that just the anolytes have big differences in the properties (see **Table 31**).

Table 31. *Active species selected for Study 3*

ID[ref]	*Class*	*D (X10^{-6} cm^2 s^{-1})[a]*	*k^0 (X10^{-6} cm s^{-1})[b]*	*E^0 (V vs NHE)[c]*
Methyl viologen[104]	*Anolyte*	*25.7*	*280*	*-0.45*
4-OH-TEMPO[104]	*Catholyte*	*29.5*	*260*	*0.80*
FcNCl[26]	*Catholyte*	*3.7*	*3.6*	*0.61*
V(II)/V(III)[74]	*Anolyte*	*2.4*	*7.0*	*-0.25*
V(IV)/V(V) [74]	*Catholyte*	*3.9*	*2.5*	*1.00*

[a] Diffusion coefficient in water at 25°C, [b] Glassy carbon electrode, [c] Approximated from half-wave potential.

Thus, despite VRFBs being the most developed RFBs – especially because of the reversible capacity loss – it is clear that alternative active species can be applied to RFBs aiming the performance optimization.

We showed that active species properties have a great influence on voltage efficiency, in the next section the most important properties are identified and discussed.

7.3 Effects of active species on voltage efficiency

Table 32 shows the values of voltage efficiency and overpotential for the 2^{5-1}-factorial design (from designed experiments of **Table 21**).

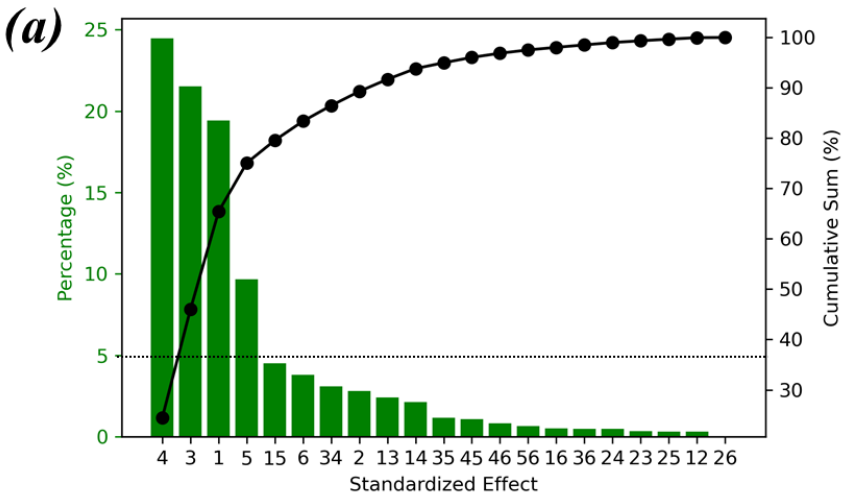
Table 32. *Responses for 2^{5-1} factorial design of six variables (Study 3)*

ID	*VE(%)*	*η^a (mV)*
1	*53.80*	*84.49*
2	*69.45*	*2.35*
3	*56.79*	*81.40*
4	*68.42*	*0.33*
5	*68.08*	*83.52*
6	*76.76*	*2.46*
7	*67.55*	*82.05*
8	*79.66*	*0.32*
9	*38.91*	*120.52*
10	*51.41*	*4.72*
11	*37.50*	*119.57*
12	*57.22*	*0.51*
13	*52.17*	*125.06*
14	*67.72*	*3.80*
15	*55.89*	*116.97*
16	*66.32*	*0.59*
17	*63.36*	*44.91*

ID	*VE(%)*	*η^a (mV)*
18	*69.82*	*1.18*
19	*63.52*	*43.99*
20	*73.56*	*0.16*
21	*72.97*	*45.06*
22	*80.16*	*1.17*
23	*75.14*	*43.88*
24	*80.15*	*0.16*
25	*45.59*	*69.45*
26	*58.16*	*1.88*
27	*49.58*	*67.22*
28	*57.94*	*0.26*
29	*61.70*	*68.91*
30	*68.01*	*1.94*
31	*61.75*	*67.59*
32	*72.03*	*0.25*

*Algorithm developed to calculate these responses is in Appendix C
[a]Concentration overpotential plus activation overpotential

Figure 65 shows the primary and secondary effects of two operating conditions – applied current density (i_{appl}) and volumetric flow (ω) – and four active species properties – standard electrochemical rate constant (k^0), diffusion coefficient (D), the concentration of active species (c_{act}) (related to solubility) and standard redox potential (E^0) – to the voltage efficiency. The most important effects (percentage > 5%) are i_{appl} (24%), E^0 (22%), k^0 (19%) and c_{act} (10%).



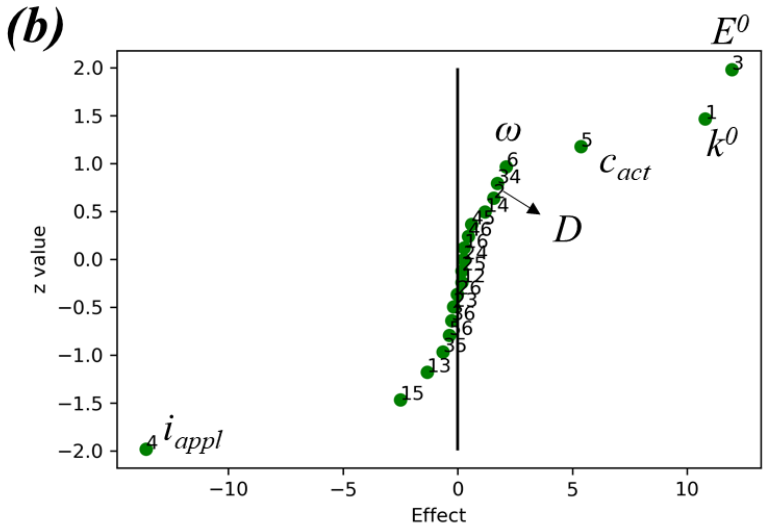


***Figure 65**. (a) Percentage of each effect on the voltage efficiency and cumulative sum and (b) the probability graph of effects*. (1) standard electrochemical rate constant (k^0), (2) Diffusion coefficient (D), (3) Standard redox potential (E^0), (4) Applied current density (i_{appl}), (5) Concentration of active species (c_{act}) and (6) volumetric flow (ω).*
**Algorithm developed to plot this graph is in Appendix C.*

Effect of applied current density on voltage efficiency. As already discussed in Study 1, the contribution of i_{appl} to decrease voltage efficiency (effect 4 = -13.58% VE) is related to the increase of ohmic and concentration overpotentials when large applied current densities are applied.

Effect of standard redox potential on voltage efficiency. If two distinct systems have the same k^0 and D and are operating at the same conditions of i_{appl}, ω and c_{act}, but have different redox potentials, the voltage efficiency is different. This conclusion arises from the effect of variable E^0 (effect 3 = 11.94% VE) going from 0.4 to 0.6 V (absolute value from both anolyte and catholyte). This is because the overpotential in the two cases is equal, but the charging and discharging Nernstian potential are different. Thus, the overpotential has a minor effect on the system with high redox potential, leading to a better voltage efficiency:

$$\begin{cases} VE_1 = \frac{\bar{E}_{d,1}-\eta}{\bar{E}_{c,1}+\eta} \\ VE_2 = \frac{\bar{E}_{d,2}-\eta}{\bar{E}_{c,2}+\eta} \end{cases} ;\ \bar{E}_{d,1} > \bar{E}_{d,2}, \bar{E}_{c,1} > \bar{E}_{c,2} \qquad \textbf{(154)}$$

Where $\bar{E}_{d,i}$ is the mean discharging Nernst potential for each system, $\bar{E}_{c,i}$ is the mean charging Nernst potential for each system and η is the total overpotential.

Thus,

$$VE_1 > VE_2 \qquad \textbf{(155)}$$

Thus, in addition to increasing the energy density and power density, higher absolute redox potentials also increase voltage efficiency.

Effect of standard electrochemical rate constant on voltage efficiency. The increase of voltage efficiency caused by the increase of k^0 (effect 1 = -10.78% VE) is related to the decrease of activation overpotential. **Figure 66** shows the activation overpotential versus the logarithm of k^0.

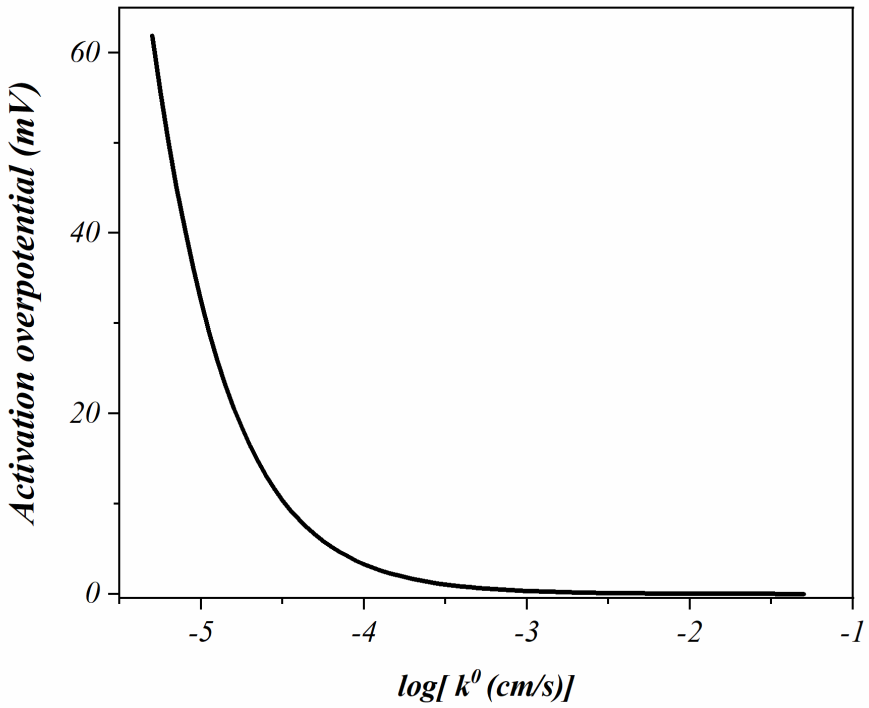


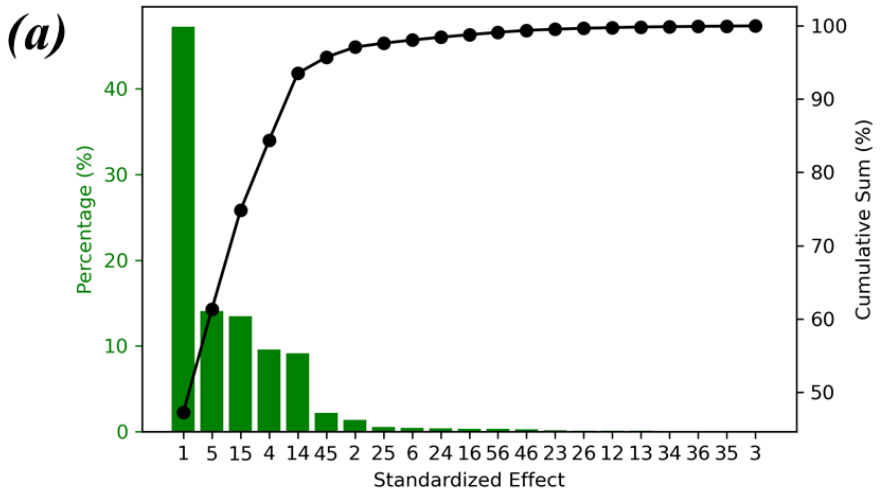
Figure 66. *Activation overpotential versus the logarithm of the standard electrochemical rate constant.*

When k^0 goes from 5X10^{-6} cm/s (logk^0 = -5.3) to 5X10^{-3} cm/s (logk^0 = -2.3), the activation overpotential decreases about 60 mV. This data shows the importance of choosing molecules that undergo facile electrochemical reactions to improve the energy efficiency. However, the behavior of the activation overpotential is not linear; thus, there is a large range of values from 1X10^{-4} cm/s ($logk^0$ = -4) to 0.03 cm/s ($logk^0$ = -1.5) where the activation overpotential is low enough.

Effect of concentration of active species on voltage efficiency. As k^0, the increase of concentration of active species decreases the activation overpotential (see **Figure 67**). The effect of both variables is a consequence of the definition of activation overpotential when $\alpha = 0.5$[95]:

$$\eta_{act}^{j} = \pm \frac{2RT}{F} sinh^{-1}\left(\frac{I}{2aFV_e k_j \sqrt{c_{ox}^{j} c_{red}^{j}}}\right) \quad \textbf{(156)}$$

Where j is the half-cell (negative or positive), a is the specific surface area of the electrode, V_e is the total volume of the electrode, I is the applied current, c_{ox} is the concentration of oxidized species and c_{red} is the concentration of reduced species in the bulk.



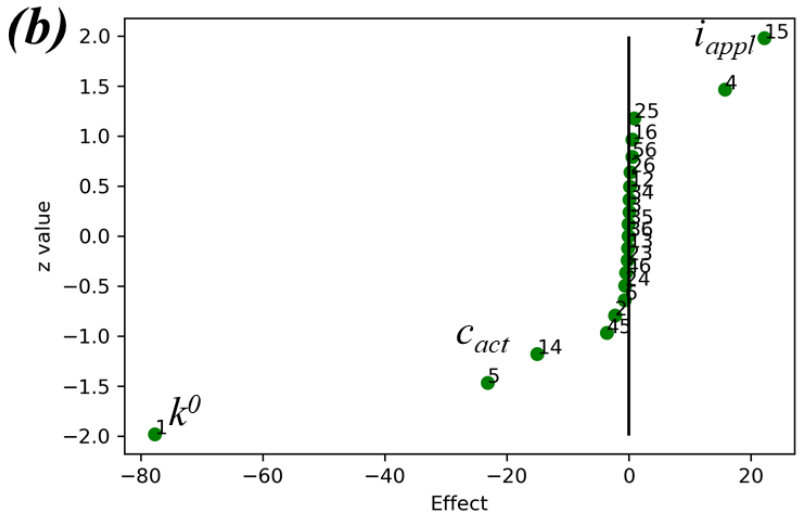


Figure 67. *(a) Percentage of each effect on the overpotential (concentration plus activation) and cumulative sum and (b) the probability graph of effects*.*
**Algorithm developed to plot this graph is in Appendix C*

Most important properties. We demonstrated that redox potential, standard electrochemical rate constant, and solubility (represented by the concentration of active species) are the most important active species properties for the improvement of voltage efficiency.

8 CONCLUSIONS

The computational approach suggested by this study provided a deep understanding of some problems related to the performance of redox flow batteries. The obtained knowledge drives us to suggest strategies and methods to mitigate the capacity loss and the voltage loss in VRFBs. Beyond that, we could understand in more detail the effects of choosing active species on the RFBs' performance.

For Study 1, we demonstrated the effects of geometric parameters in voltage loss in VRFBs. The most important variables are the inlet electrode thickness and the trapezoidal parameter L_{out}/L_{in}. We also showed that interaction effects between geometric parameters and operating conditions are important to voltage efficiency. Because of this, the optimal geometric condition depends on applied current densities. Since it is impossible to change the geometry of the cell for each applied current density, we propose a method to calculate the overall optimal geometry. For the studied scale, this geometry is rectangular and the optimal electrode thickness is about 4.5 mm.

For Study 2, we demonstrated that the concentration of active species and the applied current density are the most important variables that affect the capacity loss in VRFBs. Based on this understanding, we could provide a robust mitigation strategy based on the volume transfer between tanks in the reverse direction of net cross-contamination. The proposed strategy was successful in the mitigation of capacity loss in several combinations of applied current density and concentration, despite the capacity loss was not null at some conditions.

Therefore, we provide a set of theoretical backgrounds for experimentalists to understand the effects of several variables in the performance of redox flow batteries. We acknowledge that the proposed approach will be useful, with further experimental validation of the obtained results, for the determination of (i) optimal geometric design for any scale and type of RFB and different flow-designs, aiming at the minimization of operational costs; (ii) the bests conditions to mitigate the capacity loss in VRFBs for any scale, aiming the increasing of operational life.

For Study 3, we could provide a systematic understanding of how the choice of active species affects the energy efficiency of redox flow batteries, and not just the capacity, energy density, and power density. The most important variables were the concentration

of active species (related to the solubility), the standard electrochemical rate constant, and the redox potential. Based on this, we provide (i) useful information for experimentalists in the design of new active species and (ii) a multi-physical model capable of simulating different combinations of anolyte/catholyte to obtain performance metrics.

References

(1) Gür, T. M. Review of Electrical Energy Storage Technologies, Materials and Systems: Challenges and Prospects for Large-Scale Grid Storage. *Energy and Environmental Science*. Royal Society of Chemistry October 1, 2018, pp 2696–2767. https://doi.org/10.1039/c8ee01419a.

(2) Zhang, H.; Li, X.; Zhang, J. *Redox Flow Batteries: Fundamentals and Applications*; 2018.

(3) Hameer, S.; van Niekerk, J. L. A Review of Large-Scale Electrical Energy Storage. *International Journal of Energy Research*. John Wiley and Sons Ltd July 1, 2015, pp 1179–1195. https://doi.org/10.1002/er.3294.

(4) Khan, N.; Dilshad, S.; Khalid, R.; Kalair, A. R.; Abas, N. Review of Energy Storage and Transportation of Energy. *Energy Storage* **2019**, *1* (3), e49. https://doi.org/10.1002/est2.49.

(5) Global Energy Storage Database | Energy Storage Systems https://www.sandia.gov/ess-ssl/global-energy-storage-database/ (accessed 2021 -06 -22).

(6) Budt, M.; Wolf, D.; Span, R.; Yan, J. A Review on Compressed Air Energy Storage: Basic Principles, Past Milestones and Recent Developments. *Applied Energy*. Elsevier Ltd May 15, 2016, pp 250–268. https://doi.org/10.1016/j.apenergy.2016.02.108.

(7) Arani, A. A. K.; Karami, H.; Gharehpetian, G. B.; Hejazi, M. S. A. Review of Flywheel Energy Storage Systems Structures and Applications in Power Systems and Microgrids. *Renewable and Sustainable Energy Reviews*. Elsevier Ltd March 1, 2017, pp 9–18. https://doi.org/10.1016/j.rser.2016.11.166.

(8) Li, L.; Wei, X.; Li, B.; Yang, Z.; Luo, Q.; Wang, W. Recent Progress in Redox Flow Battery Research and Development. *Advanced Functional Materials* **2012**, *23* (8), 970–986. https://doi.org/10.1002/adfm.201200694.

(9) Wang, Y.; Liu, B.; Li, Q.; Cartmell, S.; Ferrara, S.; Deng, Z. D.; Xiao, J. Lithium and Lithium Ion Batteries for Applications in Microelectronic Devices: A Review. *Journal of Power Sources*. Elsevier July 15, 2015, pp 330–345. https://doi.org/10.1016/j.jpowsour.2015.03.164.

(10) Scrosati, B.; Garche, J. Lithium Batteries: Status, Prospects and Future. *Journal of Power Sources*. Elsevier May 1, 2010, pp 2419–2430. https://doi.org/10.1016/j.jpowsour.2009.11.048.

(11) Lourenssen, K.; Williams, J.; Ahmadpour, F.; Clemmer, R.; Tasnim, S. Vanadium Redox Flow Batteries: A Comprehensive Review. *Journal of Energy Storage* **2019**, *25*, 100844. https://doi.org/10.1016/J.EST.2019.100844.

(12) Thaller, L. H. U . S . Patent Dec . 16 , 1986, August 22, 1986.

(13) Skyllas-Kazacos, M.; Rychick, M.; Robins, R. All-Vanadium Redox Battery, January 12, 1988.

(14) Rodby, K. E.; Carney, T. J.; Ashraf Gandomi, Y.; Barton, J. L.; Darling, R. M.; Brushett, F. R. Assessing the Levelized Cost of Vanadium Redox Flow Batteries with Capacity Fade and Rebalancing. *Journal of Power Sources* **2020**, *460*, 227958. https://doi.org/10.1016/j.jpowsour.2020.227958.

(15) Ma, K.; Zhang, Y.; Liu, L.; Xi, J.; Qiu, X.; Guan, T.; He, Y. In Situ Mapping of Activity Distribution and Oxygen Evolution Reaction in Vanadium Flow Batteries. *Nature Communications* **2019**, *10* (1), 1–11. https://doi.org/10.1038/s41467-019-13147-9.

(16) Fetyan, A.; El-Nagar, G. A.; Lauermann, I.; Schnucklake, M.; Schneider, J.; Roth, C. Detrimental Role of Hydrogen Evolution and Its Temperature-Dependent Impact on the Performance of Vanadium Redox Flow Batteries. *Journal of Energy Chemistry* **2019**, *32*, 57–62. https://doi.org/10.1016/j.jechem.2018.06.010.

(17) Schweiss, R.; Pritzl, A.; Meiser, C. Parasitic Hydrogen Evolution at Different Carbon Fiber Electrodes in Vanadium Redox Flow Batteries. *Journal of The Electrochemical Society* **2016**, *163* (9), A2089–A2094. https://doi.org/10.1149/2.1281609jes.

(18) Noack, J.; Wietschel, L.; Roznyatovskaya, N.; Pinkwart, K.; Tübke, J. Techno-Economic Modeling and Analysis of Redox Flow Battery Systems. *Energies 2016, Vol. 9, Page 627* **2016**, *9* (8), 627. https://doi.org/10.3390/EN9080627.

(19) Viswanathan, V.; Crawford, A.; Stephenson, D.; Kim, S.; Wang, W.; Li, B.; Coffey, G.; Thomsen, E.; Graff, G.; Balducci, P.; Kintner-Meyer, M.; Sprenkle, V. Cost and Performance Model for Redox Flow Batteries. *Journal of Power Sources* **2014**, *247*, 1040–1051. https://doi.org/10.1016/J.JPOWSOUR.2012.12.023.

(20) Park, M.; Ryu, J.; Wang, W.; Cho, J. Material Design and Engineering of Next-Generation Flow-Battery Technologies. *Nature Reviews Materials 2016 2:1* **2016**, *2* (1), 1–18. https://doi.org/10.1038/natrevmats.2016.80.

(21) Sánchez-Díez, E.; Ventosa, E.; Guarnieri, M.; Trovò, A.; Flox, C.; Marcilla, R.; Soavi, F.; Mazur, P.; Aranzabe, E.; Ferret, R. Redox Flow Batteries: Status and Perspective towards Sustainable Stationary Energy Storage. *Journal of Power Sources* **2021**, *481*, 228804. https://doi.org/10.1016/j.jpowsour.2020.228804.

(22) Winsberg, J.; Hagemann, T.; Janoschka, T.; Hager, M. D.; Schubert, U. S. Redox-Flow Batteries: From Metals to Organic Redox-Active Materials. *Angewandte Chemie - International Edition*. John Wiley & Sons, Ltd January 16, 2017, pp 686–711. https://doi.org/10.1002/anie.201604925.

(23) Gong, K.; Fang, Q.; Gu, S.; Li, S. F. Y.; Yan, Y. Nonaqueous Redox-Flow Batteries: Organic Solvents, Supporting Electrolytes, and Redox Pairs. *Energy & Environmental Science* **2015**, *8* (12), 3515–3530. https://doi.org/10.1039/C5EE02341F.

(24) Al-Fetlawi, H.; Shah, A. A.; Walsh, F. C. Modelling the Effects of Oxygen Evolution in the All-Vanadium Redox Flow Battery. *Electrochimica Acta* **2010**, *55* (9), 3192–3205. https://doi.org/10.1016/j.electacta.2009.12.085.

(25) Gracia, R. L.; Beh, E. S.; Aziz, M. J.; Gordon, R. G.; Xia, K. T.; De Porcellinis, D. A Neutral PH Aqueous Organic–Organometallic Redox Flow Battery with Extremely High Capacity Retention. *ACS Energy Letters* **2017**, *2* (3), 639–644. https://doi.org/10.1021/acsenergylett.7b00019.

(26) Hu, B.; DeBruler, C.; Rhodes, Z.; Liu, T. L. Long-Cycling Aqueous Organic Redox Flow Battery (AORFB) toward Sustainable and Safe Energy Storage. *Journal of the American Chemical Society* **2017**, *139* (3), 1207–1214. https://doi.org/10.1021/jacs.6b10984.

(27) Matsuda, Y.; Tanaka, K.; Okada, M.; Takasu, Y.; Morita, M.; Matsumura-Inoue, T. A Rechargeable Redox Battery Utilizing Ruthenium Complexes with Non-Aqueous Organic Electrolyte. *Journal of Applied Electrochemistry 1988 18:6* **1988**, *18* (6), 909–914. https://doi.org/10.1007/BF01016050.

(28) Burnea, F. K. B.; Shi, H.; Ko, K. C.; Lee, J. Y. Reduction Potential Tuning of First Row Transition Metal MIII/MII (M = Cr, Mn, Fe, Co, Ni) Hexadentate Complexes for Viable Aqueous Redox Flow Battery Catholytes: A DFT Study. *Electrochimica Acta* **2017**, *246*, 156–164. https://doi.org/10.1016/j.electacta.2017.05.199.

(29) Luo, J.; Hu, B.; Hu, M.; Zhao, Y.; Liu, T. L. Status and Prospects of Organic Redox Flow Batteries toward Sustainable Energy Storage. *ACS Energy Letters*. September 13, 2019, pp 2220–2240. https://doi.org/10.1021/acsenergylett.9b01332.

(30) Sevov, C. S.; Hickey, D. P.; Cook, M. E.; Robinson, S. G.; Barnett, S.; Minteer, S. D.; Sigman, M. S.; Sanford, M. S. Physical Organic Approach to Persistent, Cyclable, Low-Potential Electrolytes for Flow Battery Applications. *Journal of the American Chemical Society* **2017**, *139* (8), 2924–2927. https://doi.org/10.1021/jacs.7b00147.

(31) Wei, X.; Xu, W.; Huang, J.; Zhang, L.; Walter, E.; Lawrence, C.; Vijayakumar, M.; Henderson, W. A.; Liu, T.; Cosimbescu, L.; Li, B.; Sprenkle, V.; Wang, W. Radical Compatibility with Nonaqueous Electrolytes and Its Impact on an All-Organic Redox Flow Battery. *Angewandte Chemie International Edition* **2015**, *54* (30), 8684–8687. https://doi.org/10.1002/anie.201501443.

(32) Kowalski, J. A.; Casselman, M. D.; Kaur, A. P.; Milshtein, J. D.; Elliott, C. F.; Modekrutti, S.; Attanayake, N. H.; Zhang, N.; Parkin, S. R.; Risko, C.; Brushett, F. R.; Odom, S. A. A Stable Two-Electron-Donating Phenothiazine for Application in Nonaqueous Redox Flow Batteries. *Journal of Materials Chemistry A* **2017**, *5* (46), 24371–24379. https://doi.org/10.1039/c7ta05883g.

(33) Hu, B.; Luo, J.; Hu, M.; Yuan, B.; Liu, T. L. A PH-Neutral, Metal-Free Aqueous Organic Redox Flow Battery Employing an Ammonium Anthraquinone Anolyte. *Angewandte Chemie International Edition* **2019**, *58* (46), 16629–16636. https://doi.org/10.1002/anie.201907934.

(34) Wang, H.; Sayed, S. Y.; Luber, E. J.; Olsen, B. C.; Shirurkar, S. M.; Venkatakrishnan, S.; Tefashe, U. M.; Farquhar, A. K.; Smotkin, E. S.; McCreery, R. L.; Buriak, J. M. Redox Flow Batteries: How to Determine Electrochemical Kinetic Parameters. *ACS Nano* **2020**, *14* (3), 2575–2584. https://doi.org/10.1021/acsnano.0c01281.

(35) Gautam, M.; Bhat, Z. M.; Raafik, A.; Le Vot, S.; Devendrachari, M. C.; Kottaichamy, A. R.; Dargily, N. C.; Thimmappa, R.; Fontaine, O.; Thotiyl, M. O. Coulombic Force Gated Molecular Transport in Redox Flow Batteries. *Journal of Physical Chemistry Letters* **2021**, *12* (5), 1374–1383. https://doi.org/10.1021/acs.jpclett.0c03584.

(36) Bard, A. J.; Faulkner, L. R. *Electrochemical Methods: Fundamentals and Applications*, Second Edi.; John Wiley & Sons, Inc.: New York, 2000.

(37) Cheng, D.; Li, Y.; Zhang, J.; Tian, M.; Wang, B.; He, Z.; Dai, L.; Wang, L. Recent Advances in Electrospun Carbon Fiber Electrode for Vanadium Redox Flow Battery: Properties, Structures, and Perspectives. *Carbon* **2020**, *170*, 527–542. https://doi.org/10.1016/J.CARBON.2020.08.058.

(38) Shi, Y.; Eze, C.; Xiong, B.; He, W.; Zhang, H.; Lim, T. M.; Ukil, A.; Zhao, J. Recent Development of Membrane for Vanadium Redox Flow Battery Applications: A Review. *Applied Energy* **2019**, *238* (January 2019), 202–224. https://doi.org/10.1016/j.apenergy.2018.12.087.

(39) Ke, X.; Prahl, J. M.; Alexander, J. I. D.; Wainright, J. S.; Zawodzinski, T. A.; Savinell, R. F. Rechargeable Redox Flow Batteries: Flow Fields, Stacks and Design Considerations. *Chemical Society Reviews* **2018**, *47* (23), 8721–8743. https://doi.org/10.1039/C8CS00072G.

(40) Darling, R. M.; Perry, M. L. The Influence of Electrode and Channel Configurations on Flow Battery Performance. *Journal of The Electrochemical Society* **2014**, *161* (9), A1381–A1387. https://doi.org/10.1149/2.0941409jes.

(41) Esan, O. C.; Shi, X.; Pan, Z.; Huo, X.; An, L.; Zhao, T. S. Modeling and Simulation of Flow Batteries. *Advanced Energy Materials* **2020**, 2000758. https://doi.org/10.1002/aenm.202000758.

(42) Xiong, B.; Zhao, J.; Tseng, K. J.; Skyllas-Kazacos, M.; Lim, T. M.; Zhang, Y. Thermal Hydraulic Behavior and Efficiency Analysis of an All-Vanadium Redox Flow Battery. *Journal of Power Sources* **2013**, *242*, 314–324. https://doi.org/10.1016/j.jpowsour.2013.05.092.

(43) Tang, A.; Bao, J.; Skyllas-Kazacos, M. Dynamic Modelling of the Effects of Ion Diffusion and Side Reactions on the Capacity Loss for Vanadium Redox Flow Battery. *Journal of Power Sources* **2011**, *196* (24), 10737–10747. https://doi.org/10.1016/j.jpowsour.2011.09.003.

(44) Luo, Q.; Li, L.; Nie, Z.; Wang, W.; Wei, X.; Li, B.; Chen, B.; Yang, Z. In-Situ Investigation of Vanadium Ion Transport in Redox Flow Battery. *Journal of Power Sources* **2012**, *218*, 15–20. https://doi.org/10.1016/j.jpowsour.2012.06.066.

(45) Luo, Q.; Li, L.; Wang, W.; Nie, Z.; Wei, X.; Li, B.; Chen, B.; Yang, Z.; Sprenkle, V. Capacity Decay and Remediation of Nafion-Based All-Vanadium Redox Flow

Batteries. *ChemSusChem* **2013**, *6* (2), 268–274. https://doi.org/10.1002/cssc.201200730.

(46) Wang, W.; Wei, X.; Choi, D.; Lu, X.; Yang, G.; Sun, C. Electrochemical Cells for Medium- and Large-Scale Energy Storage. In *Advances in Batteries for Medium and Large-Scale Energy Storage*; Elsevier, 2015; pp 3–28. https://doi.org/10.1016/B978-1-78242-013-2.00001-7.

(47) Li, X.; Zhang, H.; Mai, Z.; Zhang, H.; Vankelecom, I. Ion Exchange Membranes for Vanadium Redox Flow Battery (VRB) Applications. *Energy & Environmental Science* **2011**, *4* (4), 1147. https://doi.org/10.1039/c0ee00770f.

(48) Sun, C.; Chen, J.; Zhang, H.; Han, X.; Luo, Q. Investigations on Transfer of Water and Vanadium Ions across Nafion Membrane in an Operating Vanadium Redox Flow Battery. *Journal of Power Sources* **2010**, *195* (3), 890–897. https://doi.org/10.1016/j.jpowsour.2009.08.041.

(49) Maurya, S.; Shin, S.-H.; Kim, Y.; Moon, S.-H. A Review on Recent Developments of Anion Exchange Membranes for Fuel Cells and Redox Flow Batteries. *RSC Advances* **2015**, *5* (47), 37206–37230. https://doi.org/10.1039/C5RA04741B.

(50) Zhao, Y.; Yuan, Z.; Lu, W.; Li, X.; Zhang, H. The Porous Membrane with Tunable Performance for Vanadium Flow Battery: The Effect of Charge. *Journal of Power Sources* **2017**, *342*, 327–334. https://doi.org/10.1016/J.JPOWSOUR.2016.12.058.

(51) Suárez, D. J.; González, Z.; Blanco, C.; Granda, M.; Menéndez, R.; Santamaría, R. Graphite Felt Modified with Bismuth Nanoparticles as Negative Electrode in a Vanadium Redox Flow Battery. *ChemSusChem* **2014**, *7* (3), 914–918. https://doi.org/10.1002/CSSC.201301045.

(52) Wu, X. W.; Yamamura, T.; Ohta, S.; Zhang, Q. X.; Lv, F. C.; Liu, C. M.; Shirasaki, K.; Satoh, I.; Shikama, T.; Lu, D.; Liu, S. Q. Acceleration of the Redox Kinetics of VO2+/VO2 + and V3+/V2+ Couples on Carbon Paper. *Journal of Applied Electrochemistry 2011 41:10* **2011**, *41* (10), 1183–1190. https://doi.org/10.1007/S10800-011-0343-7.

(53) Gencten, M.; Sahin, Y. A Critical Review on Progress of the Electrode Materials of Vanadium Redox Flow Battery. *International Journal of Energy Research* **2020**, *44* (10), 7903–7923. https://doi.org/10.1002/ER.5487.

(54) Jirabovornwisut, T.; Kheawhom, S.; Chen, Y.-S.; Arpornwichanop, A. Optimal Operational Strategy for a Vanadium Redox Flow Battery. *Computers & Chemical*

Engineering **2020**, *136*, 106805. https://doi.org/10.1016/j.compchemeng.2020.106805.

(55) Wang, T.; Fu, J.; Zheng, M.; Yu, Z. Dynamic Control Strategy for the Electrolyte Flow Rate of Vanadium Redox Flow Batteries. *Applied Energy* **2018**, *227*, 613–623. https://doi.org/10.1016/j.apenergy.2017.07.065.

(56) Tang, A.; Bao, J.; Skyllas-Kazacos, M. Studies on Pressure Losses and Flow Rate Optimization in Vanadium Redox Flow Battery. *Journal of Power Sources* **2014**, *248*, 154–162. https://doi.org/10.1016/j.jpowsour.2013.09.071.

(57) HAN, S.; TAN, L. Thermal and Efficiency Improvements of All Vanadium Redox Flow Battery with Novel Main-Side-Tank System and Slow Pump Shutdown. *Journal of Energy Storage* **2020**, *28*, 101274. https://doi.org/10.1016/j.est.2020.101274.

(58) Pugach, M.; Parsegov, S.; Gryazina, E.; Bischi, A. Output Feedback Control of Electrolyte Flow Rate for Vanadium Redox Flow Batteries. *Journal of Power Sources* **2020**, *455*, 227916. https://doi.org/10.1016/j.jpowsour.2020.227916.

(59) Fu, J.; Wang, T.; Wang, X.; Sun, J.; Zheng, M. Dynamic Flow Rate Control for Vanadium Redox Flow Batteries. In *Energy Procedia*; Elsevier Ltd, 2017; Vol. 105, pp 4482–4491. https://doi.org/10.1016/j.egypro.2017.03.952.

(60) Xiao, W.; Tan, L. Control Strategy Optimization of Electrolyte Flow Rate for All Vanadium Redox Flow Battery with Consideration of Pump. *Renewable Energy* **2019**, *133*, 1445–1454. https://doi.org/10.1016/j.renene.2018.09.018.

(61) Fu, J.; Zheng, M.; Wang, X.; Sun, J.; Wang, T. Flow-Rate Optimization and Economic Analysis of Vanadium Redox Flow Batteries in a Load-Shifting Application. *Journal of Energy Engineering* **2017**, *143* (6), 04017064. https://doi.org/10.1061/(ASCE)EY.1943-7897.0000493.

(62) Akuzum, B.; Alparslan, Y. C.; Robinson, N. C.; Agar, E.; Kumbur, E. C. Obstructed Flow Field Designs for Improved Performance in Vanadium Redox Flow Batteries. *Journal of Applied Electrochemistry* **2019**, *49* (6), 551–561. https://doi.org/10.1007/s10800-019-01306-1.

(63) Dennison, C. R.; Agar, E.; Akuzum, B.; Kumbur, E. C. Enhancing Mass Transport in Redox Flow Batteries by Tailoring Flow Field and Electrode Design. *Journal of The Electrochemical Society* **2016**, *163* (1), A5163–A5169. https://doi.org/10.1149/2.0231601jes.

(64) Lisboa, K. M.; Marschewski, J.; Ebejer, N.; Ruch, P.; Cotta, R. M.; Michel, B.; Poulikakos, D. Mass Transport Enhancement in Redox Flow Batteries with Corrugated Fluidic Networks. *Journal of Power Sources* **2017**, *359*, 322–331. https://doi.org/10.1016/j.jpowsour.2017.05.038.

(65) Kumar, S.; Jayanti, S. Effect of Flow Field on the Performance of an All-Vanadium Redox Flow Battery. *Journal of Power Sources* **2016**, *307*, 782–787. https://doi.org/10.1016/j.jpowsour.2016.01.048.

(66) Yue, M.; Zheng, Q.; Xing, F.; Zhang, H.; Li, X.; Ma, X. Flow Field Design and Optimization of High Power Density Vanadium Flow Batteries: A Novel Trapezoid Flow Battery. *AIChE Journal* **2018**, *64* (2), 782–795. https://doi.org/10.1002/aic.15959.

(67) Gurieff, N.; Cheung, C. Y.; Timchenko, V.; Menictas, C. Performance Enhancing Stack Geometry Concepts for Redox Flow Battery Systems with Flow through Electrodes. *Journal of Energy Storage* **2019**, *22*, 219–227. https://doi.org/10.1016/j.est.2019.02.014.

(68) Zheng, Q.; Xing, F.; Li, X.; Liu, T.; Lai, Q.; Ning, G.; Zhang, H. Dramatic Performance Gains of a Novel Circular Vanadium Flow Battery. *Journal of Power Sources* **2015**, *277*, 104–109. https://doi.org/10.1016/j.jpowsour.2014.11.142.

(69) Park, S.-K.; Shim, J.; Yang, J. H.; Jin, C.-S.; Lee, B. S.; Lee, Y.-S.; Shin, K.-H.; Jeon, J.-D. The Influence of Compressed Carbon Felt Electrodes on the Performance of a Vanadium Redox Flow Battery. *Electrochimica Acta* **2014**, *116*, 447–452. https://doi.org/10.1016/j.electacta.2013.11.073.

(70) Yang, X. G.; Ye, Q.; Cheng, P.; Zhao, T. S. Effects of the Electric Field on Ion Crossover in Vanadium Redox Flow Batteries. *Applied Energy* **2015**, *145*, 306–319. https://doi.org/10.1016/j.apenergy.2015.02.038.

(71) Agar, E.; Knehr, K. W.; Chen, D.; Hickner, M. A.; Kumbur, E. C. Species Transport Mechanisms Governing Capacity Loss in Vanadium Flow Batteries: Comparing Nafion® and Sulfonated Radel Membranes. *Electrochimica Acta* **2013**, *98*, 66–74. https://doi.org/10.1016/j.electacta.2013.03.030.

(72) Song, Y.; Li, X.; Yan, C.; Tang, A. Unraveling the Viscosity Impact on Volumetric Transfer in Redox Flow Batteries. *Journal of Power Sources* **2020**, *456*, 228004. https://doi.org/10.1016/j.jpowsour.2020.228004.

(73) Li, X.; Xiong, J.; Tang, A.; Qin, Y.; Liu, J.; Yan, C. Investigation of the Use of Electrolyte Viscosity for Online State-of-Charge Monitoring Design in Vanadium

Redox Flow Battery. *Applied Energy* **2018**, *211*, 1050–1059. https://doi.org/10.1016/j.apenergy.2017.12.009.

(74) Knehr, K. W.; Agar, E.; Dennison, C. R.; Kalidindi, A. R.; Kumbur, E. C. A Transient Vanadium Flow Battery Model Incorporating Vanadium Crossover and Water Transport through the Membrane. *Journal of The Electrochemical Society* **2012**, *159* (9), A1446–A1459. https://doi.org/10.1149/2.017209jes.

(75) Agar, E.; Benjamin, A.; Dennison, C. R.; Chen, D.; Hickner, M. A.; Kumbur, E. C. Reducing Capacity Fade in Vanadium Redox Flow Batteries by Altering Charging and Discharging Currents. *Journal of Power Sources* **2014**, *246*, 767–774. https://doi.org/10.1016/j.jpowsour.2013.08.023.

(76) Lu, M. Y.; Yang, W. W.; Deng, Y. M.; Li, W. Z.; Xu, Q.; He, Y. L. Mitigating Capacity Decay and Improving Charge-Discharge Performance of a Vanadium Redox Flow Battery with Asymmetric Operating Conditions. *Electrochimica Acta* **2019**, *309*, 283–299. https://doi.org/10.1016/j.electacta.2019.04.032.

(77) Park, J. H.; Park, J. J.; Park, O. O.; Yang, J. H. Capacity Decay Mitigation by Asymmetric Positive/Negative Electrolyte Volumes in Vanadium Redox Flow Batteries. *ChemSusChem* **2016**, *9* (22), 3181–3187. https://doi.org/10.1002/cssc.201601110.

(78) Bhattarai, A.; Wai, N.; Schweiss, R.; Whitehead, A.; Scherer, G. G.; Ghimire, P. C.; Lim, T. M.; Hng, H. H. Vanadium Redox Flow Battery with Slotted Porous Electrodes and Automatic Rebalancing Demonstrated on a 1 kW System Level. *Applied Energy* **2019**, *236*, 437–443. https://doi.org/10.1016/J.APENERGY.2018.12.001.

(79) Wang, K.; Liu, L.; Xi, J.; Wu, Z.; Qiu, X. Reduction of Capacity Decay in Vanadium Flow Batteries by an Electrolyte-Reflow Method. *Journal of Power Sources* **2017**, *338*, 17–25. https://doi.org/10.1016/J.JPOWSOUR.2016.11.031.

(80) Frías-Ferrer, Á.; González-García, J.; Sáez, V.; de Ponce León, C.; Walsh, F. C. The Effects of Manifold Flow on Mass Transport in Electrochemical Filter-Press Reactors. *AIChE Journal* **2008**, *54* (3), 811–823. https://doi.org/10.1002/AIC.11426.

(81) Frias-Ferrer, A.; González-García, J.; Sáez, V.; Expósito, E.; Sánchez-Sánchez, C. M.; Montiel, V.; Aldaz, A.; Walsh, F. C. The Entrance and Exit Effects in Small Electrochemical Filter-Press Reactors Used in the Laboratory. *Journal of Chemical*

Education **2005**, *82* (9), 1395–1398. https://doi.org/10.1021/ED082P1395/SUPPL_FILE/JCE2005P1395W.ZIP.

(82) Leung, P.; Li, X.; Ponce De León, C.; Berlouis, L.; Low, C. T. J.; Walsh, F. C. Progress in Redox Flow Batteries, Remaining Challenges and Their Applications in Energy Storage. *RSC Advances* **2012**, *2* (27), 10125–10156. https://doi.org/10.1039/c2ra21342g.

(83) Robb, B. H.; Waters, S. E.; Marshak, M. P. Evaluating Aqueous Flow Battery Electrolytes: A Coordinated Approach. *Dalton Transactions* **2020**, *49* (45), 16047–16053. https://doi.org/10.1039/D0DT02462G.

(84) Moaveni, S. *Finite Element Analysis: Theory and Application with ANSYS*, first.; Prentice-Hall: New Jersey, 199AD.

(85) Detailed Explanation of the Finite Element Method (FEM) https://www.comsol.com/multiphysics/finite-element-method (accessed 2021 -12 -13).

(86) Seshu, P. *Textbook of Finite Element Analysis*, first.; PHI Learning Private Limited: New Delhi, 2012.

(87) Brereton, R. G. *Chemometrics: Data Analysis for the Laboratory and Chemical Plant*, first.; Wiley, 203AD.

(88) Bruns, R. E.; Scarminio, I. S.; de Barros Neto, B. *Statistical Design - Chemometrics*, First.; Rutan, S., Walczak, B., Eds.; Elsevier B.V., 2006.

(89) Pereira Filho, E. R. *Planejamento Fatorial Em Química: Maximizando a Obtenção de Resultados*; EdUFSCar: São Carlos, 2018.

(90) Shah, A. A.; Watt-Smith, M. J.; Walsh, F. C. A Dynamic Performance Model for Redox-Flow Batteries Involving Soluble Species. *Electrochimica Acta* **2008**, *53* (27), 8087–8100. https://doi.org/10.1016/j.electacta.2008.05.067.

(91) D. A. Knopf, *; B. P. Luo; U. K. Krieger, and; Koop, T. Thermodynamic Dissociation Constant of the Bisulfate Ion from Raman and Ion Interaction Modeling Studies of Aqueous Sulfuric Acid at Low Temperatures. **2003**. https://doi.org/10.1021/JP027775+.

(92) Newman, J.; Thomas-Alyea. *Electrochemical Systems*, 3rd ed.; New Jersey, 2004.

(93) Yamamura, T.; Watanabe, N.; Yano, T.; Shiokawa, Y. Electron-Transfer Kinetics of Np[Sup 3+]/Np[Sup 4+], NpO[Sub 2][Sup +]/NpO[Sub 2][Sup 2+], V[Sup 2+]/V[Sup 3+], and VO[Sup 2+]/VO[Sub 2][Sup +] at Carbon Electrodes. *Journal*

of The Electrochemical Society **2005**, *152* (4), A830. https://doi.org/10.1149/1.1870794.

(94) Ishitobi, H.; Saito, J.; Sugawara, S.; Oba, K.; Nakagawa, N. Visualized Cell Characteristics by a Two-Dimensional Model of Vanadium Redox Flow Battery with Interdigitated Channel and Thin Active Electrode. *Electrochimica Acta* **2019**. https://doi.org/10.1016/j.electacta.2019.04.055.

(95) Murthy, S. K.; Sharma, A. K.; Choo, C.; Birgersson, E. Analysis of Concentration Overpotential in an All-Vanadium Redox Flow Battery. *Journal of The Electrochemical Society* **2018**, *165* (9), A1746–A1752. https://doi.org/10.1149/2.0681809jes.

(96) Pourcelly, G.; Lindheimer, A.; Gavach, C.; Hurwitz, H. D. Electrical Transport of Sulphuric Acid in Nation Perfluorosulphonic Membranes. *Journal of Electroanalytical Chemistry* **1991**, *305* (1), 97–113. https://doi.org/10.1016/0022-0728(91)85205-4.

(97) Verbrugge, M. W. Ion and Solvent Transport in Ion-Exchange Membranes. *Journal of The Electrochemical Society* **1990**, *137* (3), 886. https://doi.org/10.1149/1.2086573.

(98) Zaikov, G.; Iordanskii, A.; Markin, V. *Diffusion of Electrolytes in POlymers*, 1st ed.; Utrecht.

(99) Lei, Y.; Zhang, B. W.; Zhang, Z. H.; Bai, B. F.; Zhao, T. S. An Improved Model of Ion Selective Adsorption in Membrane and Its Application in Vanadium Redox Flow Batteries. *Applied Energy* **2018**, *215*, 591–601. https://doi.org/10.1016/j.apenergy.2018.02.042.

(100) Bernardi, D. M.; Verbrugge, M. W. Mathematical Model of a Gas Diffusion Electrode Bonded to a Polymer Electrolyte. *AIChE Journal* **1991**, *37* (8), 1151–1163. https://doi.org/10.1002/aic.690370805.

(101) Kim, D. K.; Yoon, S. J.; Lee, J.; Kim, S. Parametric Study and Flow Rate Optimization of All-Vanadium Redox Flow Batteries. *Applied Energy* **2018**, *228*, 891–901. https://doi.org/10.1016/j.apenergy.2018.06.094.

(102) Kim, S.; Yan, J.; Schwenzer, B.; Zhang, J.; Li, L.; Liu, J.; Yang, Z.; Hickner, M. A. Cycling Performance and Efficiency of Sulfonated Poly(Sulfone) Membranes in Vanadium Redox Flow Batteries. *Electrochemistry Communications* **2010**, *12* (11), 1650–1653. https://doi.org/10.1016/J.ELECOM.2010.09.018.

(103) Lei, Y.; Zhang, B. W.; Bai, B. F.; Zhao, T. S. A Transient Electrochemical Model Incorporating the Donnan Effect for All-Vanadium Redox Flow Batteries. *Journal of Power Sources* **2015**, *299*, 202–211. https://doi.org/10.1016/j.jpowsour.2015.08.100.

(104) Liu, T.; Wei, X.; Nie, Z.; Wang, W.; Liu, T.; Sprenkle, V. A Total Organic Aqueous Redox Flow Battery Employing a Low Cost and Sustainable Methyl Viologen Anolyte and 4-HO-TEMPO Catholyte. *Advanced Energy Materials* **2015**, *6* (3), 1501449. https://doi.org/10.1002/aenm.201501449.

(105) Tsushima, S.; Suzuki, T. Modeling and Simulation of Vanadium Redox Flow Battery with Interdigitated Flow Field for Optimizing Electrode Architecture. *Journal of The Electrochemical Society* **2020**, *167* (2), 020553. https://doi.org/10.1149/1945-7111/ab6dd0.

(106) Chen, H.; Cong, G.; Lu, Y.-C. Recent Progress in Organic Redox Flow Batteries: Active Materials, Electrolytes and Membranes. *Journal of Energy Chemistry* **2018**, *27* (5), 1304–1325. https://doi.org/10.1039/c8cc02336k.

(107) Ding, Y.; Zhang, C.; Zhang, L.; Zhou, Y.; Yu, G. Molecular Engineering of Organic Electroactive Materials for Redox Flow Batteries. *Chemical Society Reviews* **2018**, *47* (1), 69–103. https://doi.org/10.1039/C7CS00569E.

(108) Pragst, F.; Ziebig, R.; Seydewitz, U.; Driesel, G. Electrochemistry of Pyrylium Compounds—II[1]. Structural Effects on the Voltammetric Behavior in Acetonitrile. *Electrochimica Acta* **1980**, *25* (3), 341–352. https://doi.org/10.1016/0013-4686(80)90016-X.

Appendix A – Scripts for data treatment (Study 1)

Algorithm to calculate the responses from COMSOL files (Table 22)

The following algorithm (in python) was developed for the calculations of efficiencies (coulombic, voltage, energetic, and battery) based on the multi-physical model.

Input files: (COMSOL files) cell potential versus time, faradaic current versus time and pump power versus time.

```
### importing libraries
import pandas as pd
import numpy as np
from scipy.signal import find_peaks
from scipy.integrate import trapz
import matplotlib.pyplot as plt


### reading files
Vt = input('Name of file V versus time:')
it = input('Name of file i versus time:')
ppump = input('Name of file p_pump versus time:')
file_Vt = 'inputname.txt'.replace('inputname',Vt)
file_it = 'inputname.txt'.replace('inputname',it)
file_ppump = 'inputname.txt'.replace('inputname',ppump)
df1_raw = pd.read_csv(file_Vt,skiprows=(7))
df2_raw = pd.read_csv(file_it,skiprows=(7))
df3_raw = pd.read_csv(file_ppump,skiprows=(7))
df1 = df1_raw.rename(columns={'% X':'time','Height':'V'})
df2 = df2_raw.rename(columns={'% X':'time','Height':'i'})
df3 = df3_raw.rename(columns={'% X':'time','Height':'P_pump'})

### differentiation of discrete values of V versus time data (for time peak determination)
x_V = list(df1['time'])
y_V = list(df1['V'])

dydx_V=[]

for i in range(len(x_V)):
    if i==0:
        dx=x_V[i:i+2]
        dy=y_V[i:i+2]
        order=1
    elif i==len(x_V)-1:
        dx=x_V[i-1:i+1]
        dy=y_V[i-1:i+1]
        order=1
    else:
        dx=x_V[i-1:i+2]
        dy=y_V[i-1:i+2]
```

```
        order=2
    z=np.polyfit(dx,dy,len(dx)-1)
    f=np.poly1d(z)
    df=np.polyder(f)
    dydx_V.append(float(df(x_V[i])))
dydx_V=np.array(dydx_V)



### finding peaks charge/discharge
peaks = find_peaks(abs(dydx_V),height=0.000005,threshold=0.000005)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
    height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x_V)[peaks[0]]
if peak_pos.shape[0] == 1:
    peak_pos = np.append(np.array(x_V)[peaks[0]],x_V[len(x_V)-1])


### plotting peaks
fig = plt.figure()
ax = fig.subplots()
ax.plot(x_V,abs(dydx_V),'-k', label = 'Time derivative')
ax.scatter(peak_pos, height, color = 'r', s = 15, marker = 'D', label = 'Peaks')
ax.legend()
ax.grid()


### calculating capacity loss
step_time = []
step_time.append(peak_pos[0])
z_ite = 0
for i in range(len(peak_pos)):
    if z_ite == len(peak_pos)-1:
        break
    step_time.append(peak_pos[z_ite+1]-peak_pos[z_ite])
    z_ite += 1
discharge_time = step_time[1:len(step_time):2]
capacity_loss = []
for i in range(len(discharge_time)):
    capacity_loss.append(100*discharge_time[i]/discharge_time[0])


### preparing index, dataframes arrays for efficiencies calculation
discharge_steps = peak_pos[1:len(peak_pos):2]
charge_steps = peak_pos[0:len(peak_pos):2]

remove_duplicated_times_V = ~df1.duplicated(subset='time')
df1_corr = df1[remove_duplicated_times_V]
time_V = np.array(df1_corr['time'])
potential = np.array(df1_corr['V'])

remove_duplicated_times_i = ~df2.duplicated(subset='time')
df2_corr = df2[remove_duplicated_times_i]
time_i = np.array(df2_corr['time'])
current = np.array(df2_corr['i'])

remove_duplicated_times_P_pump = ~df3.duplicated(subset='time')
df3_corr = df3[remove_duplicated_times_P_pump]
time_P_pump = np.array(df3_corr['time'])
P_pump = np.array(df3_corr['P_pump'])

charge_time = step_time[0:len(step_time):2]

a = 0
index_discharge = []
```

```
while a < discharge_steps.shape[0]:
    for i in range(time_V.shape[0]):
        if time_V[i] == discharge_steps[a]:
            index_discharge.append(i)
    a += 1


b = 0
index_charge = []
while b < (charge_steps.shape[0]):
    for i in range(time_V.shape[0]):
        if time_V[i] == charge_steps[b]:
            index_charge.append(i)
    b += 1


### calculating voltage efficiency
Avg_V = []
Avg_V.append((trapz(potential[0:index_charge[0]],x=time_V[0:index_charge[0]]))/charge_time[0])

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    Avg_dis =
(trapz(potential[index_charge[e]:index_discharge[e]],x=time_V[index_charge[e]:index_discharge[e]]))/discharge_ti
me[e]
    Avg_V.append(Avg_dis)
    if x_ite == len(charge_time):
      break
    Avg_cha =
(trapz(potential[index_discharge[e]:index_charge[x_ite]],x=time_V[index_discharge[e]:index_charge[x_ite]]))/charg
e_time[x_ite]
    Avg_V.append(Avg_cha)
    e += 1
    x_ite +=1

VE_prev = []

for i in range(len(Avg_V)-1):
    h = 100*Avg_V[i+1]/Avg_V[i]
    VE_prev.append(h)

VE = VE_prev[0::2]



### calculating coulombic efficiency
Q = []

f = 0
y_ite  = 1

for i in range(n+1):
    Q_cha = current.mean()*step_time[f]
    Q.append(Q_cha)
    Q_dis = current.mean()*step_time[y_ite]
    Q.append(Q_dis)
    f += 1
    y_ite += 1

CE_prev = []

for i in range(len(Q)-1):
    g = 100*Q[i+1]/Q[i]
```

```
    CE_prev.append(g)

CE = CE_prev[0::2]



###calculating energy efficiency

EE = []
k = 0
for i in range(len(VE)):
    l = VE[k]*CE[k]/100
    EE.append(l)
    k += 1

#calculating pumping energy consumption in Joules
E_cons = []
E_cons.append(trapz(P_pump[0:index_charge[0]],x=time_P_pump[0:index_charge[0]]))

m = 0
z_ite = 1

for i in range(n+1):
    if m == len(discharge_time):
        break
    E_cons_dis =
trapz(P_pump[index_charge[m]:index_discharge[m]],x=time_P_pump[index_charge[m]:index_discharge[m]])
    E_cons.append(E_cons_dis)
    if z_ite == len(charge_time):
        break
    E_cons_cha =
trapz(P_pump[index_discharge[m]:index_charge[z_ite]],x=time_P_pump[index_discharge[m]:index_charge[z_ite]])
    E_cons.append(E_cons_cha)
    m += 1
    z_ite += 1


###calculating battery efficiency
E = []

E.append(Avg_V[0]*Q[0]-E_cons[0])
E.append(Avg_V[1]*Q[1]-E_cons[1])

n_ite = 2
o_ite = 3

for i in range(n):
    if n_ite == len(Avg_V):
        break
    E_cha = Avg_V[n_ite]*Q[n_ite]+E_cons[n_ite]
    E.append(E_cha)
    if o_ite == len(Avg_V):
        break
    E_dis = Avg_V[o_ite]*Q[o_ite]-E_cons[o_ite]
    E.append(E_dis)
    n_ite += 2
    o_ite += 2

BE_prev = []

p_ite = 0

for i in range(len(E)-1):
    q_const = 100*E[i+1]/E[i]
    BE_prev.append(q_const)

BE = BE_prev[0::2]
```

```
###creating and exporting DataFrames
n_cycles = np.arange(1,n+2,1)
capacity_loss_df = pd.DataFrame({'n_cycles':n_cycles,'capacity_loss':np.array(capacity_loss)})

efficiencies_df = pd.DataFrame({'n_cycles':n_cycles,'voltage efficiency':np.array(VE),'coulombic
efficiency':np.array(CE),'energy efficiency':np.array(EE),
                        'battery efficiency':np.array(BE)})

efficiencies_df.to_csv('efficiencies_df.txt',sep='\t',decimal=',',index=False)



###plotting graphs
fig1,ax1 = plt.subplots()
ax1.plot(n_cycles,np.array(VE),'bo',label = 'Voltage')
ax1.plot(n_cycles,np.array(CE),'ro',label = 'Coulombic')
ax1.plot(n_cycles,np.array(EE),'go',label = 'Energy')
ax1.plot(n_cycles,np.array(BE),'ko',label = 'Battery')
ax1.set(xlabel = 'Cycle number',ylabel = 'Efficiency(%)', title = 'Efficiencies')
plt.xlim(0,n)
plt.xticks(n_cycles+1)
plt.ylim(40,105)
plt.legend()
plt.savefig('efficiencies.png')

###selecting data from first cycle
cycle1 =[CE[0],VE[0],EE[0],BE[0],E_cons[0]]
```

The following algorithm (in python) was developed for the calculations of activation and concentration overpotentials based on multi-physical model output files: cell voltage, concentration overpotential and activation overpotential. The calculated value is the mean overpotential in discharging procedure.

Input files: activation overpotential versus time, concentration overpotential versus time and cell potential versus time.

```
### importing libraries
import pandas as pd
import numpy as np
from scipy.signal import find_peaks
import matplotlib.pyplot as plt


### reading files
eta_t = input('Name of file eta versus time:')
Vt = input('Name of file V versus time:')
file_eta_t = 'inputname.txt'.replace('inputname',eta_t)
file_Vt = 'inputname.txt'.replace('inputname',Vt)
df1_raw = pd.read_csv(file_Vt,skiprows=(7))
df4_raw = pd.read_csv(file_eta_t,skiprows=(7))
df1 = df1_raw.rename(columns={'% X':'time','Height':'V'})
df4 = df4_raw.rename(columns={'% X':'time','Height':'eta'})

### differentiation of discrete values of V versus time data (for time peak determination)
x_V = list(df1['time'])
y_V = list(df1['V'])

dydx_V=[]

for i in range(len(x_V)):
    if i==0:
        dx=x_V[i:i+2]
        dy=y_V[i:i+2]
        order=1
    elif i==len(x_V)-1:
        dx=x_V[i-1:i+1]
        dy=y_V[i-1:i+1]
        order=1
    else:
        dx=x_V[i-1:i+2]
        dy=y_V[i-1:i+2]
        order=2
    z=np.polyfit(dx,dy,len(dx)-1)
    f=np.poly1d(z)
    df=np.polyder(f)
    dydx_V.append(float(df(x_V[i])))
dydx_V=np.array(dydx_V)


### finding peaks charge/discharge
peaks = find_peaks(abs(dydx_V),height=0.000005,threshold=0.000005)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
    height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])
```

```
peak_pos = np.array(x_V)[peaks[0]]
if peak_pos.shape[0] == 1:
    peak_pos = np.append(np.array(x_V)[peaks[0]],x_V[len(x_V)-1])



### preparing index, dataframes arrays
discharge_steps = peak_pos[1:len(peak_pos):2]
charge_steps = peak_pos[0:len(peak_pos):2]

step_time = []
for i in range(len(peak_pos)-1):
    step_time.append(-peak_pos[i]+peak_pos[i+1])

discharge_time = step_time[1:len(step_time):2]
charge_time = step_time[0:len(step_time):2]

remove_duplicated_times_V = ~df1.duplicated(subset='time')
df1_corr = df1[remove_duplicated_times_V]
time_V = np.array(df1_corr['time'])
potential = np.array(df1_corr['V'])

remove_duplicated_times_eta = ~df4.duplicated(subset='time')
df4_corr = df4[remove_duplicated_times_eta]
time_eta = np.array(df4_corr['time'])
overpotential = np.array(df4_corr['eta'])

charge_time = step_time[0:len(step_time):2]

a = 0
index_discharge = []
while a < discharge_steps.shape[0]:
    for i in range(time_V.shape[0]):
        if time_V[i] == discharge_steps[a]:
            index_discharge.append(i)
    a += 1


b = 0
index_charge = []
while b < (charge_steps.shape[0]):
    for i in range(time_V.shape[0]):
        if time_V[i] == charge_steps[b]:
            index_charge.append(i)
    b += 1


### selecting overpotential from the first cycle
charge= pd.DataFrame({'time': time_eta[0:index_charge[0]],
                      'overpotential': overpotential[0:index_charge[0]]})
discharge= pd.DataFrame({'time': time_eta[index_charge[0]:index_discharge[0]],
                      'overpotential': overpotential[index_charge[0]:index_discharge[0]]})

###data output

eta = 1000*abs(discharge['overpotential']).mean
print(eta)
```

The following algorithm (in python) was developed for the calculations of ohmic overpotential based on multi-physical model. The calculated value is the mean overpotential in discharging procedure.

Input files: cell potential versus time and ohmic overpotential versus time.

```
### importing libraries
import pandas as pd
import numpy as np
from scipy.signal import find_peaks

### reading files
V_ohm = input('Name of file V_ohm versus time:')
Vt = input('Name of file V versus time:')
file_V_ohm = 'inputname.txt'.replace('inputname',V_ohm)
file_Vt = 'inputname.txt'.replace('inputname',Vt)
df1_raw = pd.read_csv(file_Vt,skiprows=(7))
df4_raw = pd.read_csv(file_V_ohm,skiprows=(7))
df1 = df1_raw.rename(columns={'% X':'time','Height':'V'})
df4 = df4_raw.rename(columns={'% X':'time','Height':'V_ohm'})

### differentiation of discrete values of V versus time data (for time peak determination)
x_V = list(df1['time'])
y_V = list(df1['V'])

dydx_V=[]

for i in range(len(x_V)):
    if i==0:
        dx=x_V[i:i+2]
        dy=y_V[i:i+2]
        order=1
    elif i==len(x_V)-1:
        dx=x_V[i-1:i+1]
        dy=y_V[i-1:i+1]
        order=1
    else:
        dx=x_V[i-1:i+2]
        dy=y_V[i-1:i+2]
        order=2
    z=np.polyfit(dx,dy,len(dx)-1)
    f=np.poly1d(z)
    df=np.polyder(f)
    dydx_V.append(float(df(x_V[i])))
dydx_V=np.array(dydx_V)



### finding peaks charge/discharge
peaks = find_peaks(abs(dydx_V),height=0.000005,threshold=0.000005)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
```

```
    height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x_V)[peaks[0]]
if peak_pos.shape[0] == 1:
    peak_pos = np.append(np.array(x_V)[peaks[0]],x_V[len(x_V)-1])

### preparing index, dataframes arrays
discharge_steps = peak_pos[1:len(peak_pos):2]
charge_steps = peak_pos[0:len(peak_pos):2]

step_time = []
for i in range(len(peak_pos)-1):
    step_time.append(-peak_pos[i]+peak_pos[i+1])

discharge_time = step_time[1:len(step_time):2]
charge_time = step_time[0:len(step_time):2]

remove_duplicated_times_V = ~df1.duplicated(subset='time')
df1_corr = df1[remove_duplicated_times_V]
time_V = np.array(df1_corr['time'])
potential = np.array(df1_corr['V'])

remove_duplicated_times_V_ohm = ~df4.duplicated(subset='time')
df4_corr = df4[remove_duplicated_times_V_ohm]
time_V_ohm = np.array(df4_corr['time'])
V_ohm = np.array(df4_corr['V_ohm'])

charge_time = step_time[0:len(step_time):2]

a = 0
index_discharge = []
while a < discharge_steps.shape[0]:
    for i in range(time_V.shape[0]):
        if time_V[i] == discharge_steps[a]:
            index_discharge.append(i)
    a += 1


b = 0
index_charge = []
while b < (charge_steps.shape[0]):
    for i in range(time_V.shape[0]):
        if time_V[i] == charge_steps[b]:
            index_charge.append(i)
    b += 1


### selecting ohmic drop from the first cycle
charge= pd.DataFrame({'time': time_V_ohm[0:index_charge[0]],
                      'V_ohm': V_ohm[0:index_charge[0]]})
discharge= pd.DataFrame({'time': time_V_ohm[index_charge[0]:index_discharge[0]],
                      'V_ohm': V_ohm[index_charge[0]:index_discharge[0]]})

###data output
ohm = 1000*abs(discharge['V_ohm']).mean()
print(ohm)
```

Algorithm for regression analysis (Table 23**)**

The following algorithm (in python) was developed for regression analysis. The variables codified are used. The purpose was to obtain the coefficients on the same scale for all variables.

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

```
### importing libraries
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt
import numpy as np


### reading files
file_name=str(input('File name: '))
df_prev = 'inputname.txt'.replace('inputname',file_name)
df = pd.read_csv(df_prev,sep='\t')

### selecting matrix X and vector y
X = df.iloc[:, 0:5].values
y = df.iloc[:, 5].values

### setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
regressor = LinearRegression()
regressor.fit(X1,y)

### calculating predict values and coefficients
y_prev = regressor.predict(X1)
coef = regressor.coef_
inpt = regressor.intercept_

### dettermining metrics
RMSE = mean_squared_error(y, y_prev,squared=False)
r2 = r2_score(y,y_prev)
r2s = str(round(r2,4))
RMSEs = str(round(RMSE,2))
fig1, ax1 = plt.subplots()
ax1.plot(np.linspace(round(y.min()),round(y.max()),200),np.linspace(round(y.min()),round(y.max()),200),'k--')
ax1.plot(y,y_prev,'bo',markersize=7)



### ploting and exporting graphs
answer_name = str(input('Answer name (unit): '))
ax1.set(xlabel = 'Simulated '+ answer_name,ylabel = 'Predicted '+ answer_name)
plt.grid()
plt.text(70,50,' $R^2$:' + r2s + '\n RMSE:' + RMSEs)

plt.savefig('Regression_'+file_name+'.tiff',format='tiff',dpi=600)
```

Algorithm to calculate the statistical significance of coefficients (Table 24 **and Table** 25**)**

The following algorithm (in python) was developed for the determination of statistical significance.

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

```
### importing libraries
import numpy as np
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from scipy import stats

### reading data
file_name=str(input('File name input: '))
df_prev = 'inputname.txt'.replace('inputname',file_name)
df = pd.read_csv(df_prev,sep='\t')
df2 = pd.read_csv('coef.txt',sep='\t')

### selecting matrix X and vector y
X = df2.iloc[:, 0:5].values
y = df.iloc[:, 5].values

### setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
regressor = LinearRegression()
regressor.fit(X1,y)

### calculating coefficients
y_prev = regressor.predict(X1)
coef = regressor.coef_
coef[0] = regressor.intercept_


w0,w1,w2,w3,w4,w5 = coef[0],coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]

### calculating confidence intervals for coefficients
MSE = mean_squared_error(y, y_prev,squared=True)
A = MSE*np.linalg.inv(np.dot(X1.T,X1)).diagonal()
sqrtA = np.sqrt(A)
t = stats.t.ppf(1-0.05/2,(X.shape[0]-len(coef)-1))
ci = sqrtA*t


### selecting significant coefficients

coef_plus = coef+ci
coef_minus = coef-ci
```

```
a_list = []
b_list = []

for i in range(len(coef_plus)):
    c_const = coef_plus[i]>0
    a_list.append(c_const)

for i in range(len(coef_minus)):
    d_const = coef_minus[i]>0
    b_list.append(d_const)

e_list = []

for i in range(len(a_list)):
    if a_list[i] == b_list[i]:
        f_const = 'yes'
        e_list.append(f_const)
    else:
        g_const = ' '
        e_list.append(g_const)


### creating and exporting dataframe
coef_name = np.array(['w0','w01','w02','w03','w04','w05',
                      'w11','w12','w13','w14','w15',
                      'w22','w23','w24','w25',
                      'w33','w34','w35',
                      'w44','w45',
                      'w55'])
coefsig_df = pd.DataFrame({'coefficient':coef_name,'significant?':np.array(e_list)})


coefsig_df.to_csv(file_name+'_coef.txt',sep='\t',decimal=',',index=False)
```

Algorithm to plot Figure 38

The following algorithm (in python) was developed for regression analysis and 4D plots. The variables decodified were used.

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

bB corresponds to L_{out}/L;

i corresponds to i_{appl};

Q correspond to ω.

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt
import numpy as np
import matplotlib as mpl
from matplotlib import cm

# reading data
df_prev = 'VE.txt'
df = pd.read_csv(df_prev,sep='\t')

# selecting matrix X and vector y
X = df.iloc[:, 0:5].values
y = df.iloc[:, 5].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
regressor = LinearRegression()
regressor.fit(X1,y)

# calculating predict values and coefficients
y_prev = regressor.predict(X1)
coef = regressor.coef_

#dettermining metrics
RMSE = mean_squared_error(y, y_prev,squared=False)
r2 = r2_score(y,y_prev)
r2s = str(round(r2,4))
RMSEs = str(round(RMSE,2))

answer_name = 'Normalized VE (%)'

# preparing data for 4D plots
levels = 300
points = 200
transp = 0.9

i1 = 40
i2 = 100
i3 = 160
```

```
bB1 = 0.5
bB2 = 1
bB3 = 1.5
Q1 = 0.15
Q2 = 0.5
Q3 = 0.85
L1 = 2.5
L2 = 5
L3 = 7.5
H1 = 14.5
H2 = 30
H3 = 45.5

L = np.linspace(L1,L3,points)
H = np.linspace(H1,H3,points)
bB = np.linspace(bB1,bB3,points)

w = regressor.intercept_
w0,w1,w2,w3,w4,w5 = coef[0],coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]

# L versus bB versus current
const7 =
w+w0+w2*i1+w3*Q2+w4*H2+w22*(i1**2)+w23*i1*Q2+w24*i1*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)
const8 =
w+w0+w2*i2+w3*Q2+w4*H2+w22*(i2**2)+w23*i2*Q2+w24*i2*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)
const9 =
w+w0+w2*i3+w3*Q2+w4*H2+w22*(i3**2)+w23*i3*Q2+w24*i3*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)

del L
del H
del bB

L = np.linspace(L1,L3,points)
H = np.linspace(H1,H3,points)
bB = np.linspace(bB1,bB3,points)

L, bB   = np.meshgrid(L, bB)
Z7 =
const7+w1*L+w5*bB+w11*(L**2)+w12*L*i1+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i1*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)
Z8 =
const8+w1*L+w5*bB+w11*(L**2)+w12*L*i2+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i2*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)
Z9 =
const9+w1*L+w5*bB+w11*(L**2)+w12*L*i3+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i3*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)

Z7n=Z7/Z7.max()
Z8n=Z8/Z8.max()
Z9n=Z9/Z9.max()

ZQ_min = np.array([round(Z7n.min(),2),round(Z8n.min(),2),round(Z9n.min(),2)]).min()
ZQ_max = np.array([round(Z7n.max(),2),round(Z8n.max(),2),round(Z9n.max())]).max()

fig4 = plt.figure()
ax4 = fig4.gca(projection='3d')

plt.contourf(L, bB, Z7n, levels = levels,alpha=transp,offset=i1)
plt.clim(ZQ_min,ZQ_max)

plt.contourf(L, bB, Z8n, levels = levels,alpha=transp,offset=i2)
plt.clim(ZQ_min,ZQ_max)
```

```
plt.contourf(L, bB, Z9n, levels = levels,alpha=transp,offset=i3)
plt.clim(ZQ_min,ZQ_max)

norm = mpl.colors.Normalize(vmin=ZQ_min,vmax=ZQ_max)
cb4=plt.colorbar(cm.ScalarMappable(norm=norm),
                 orientation='vertical',
                 ticks=np.arange(ZQ_min,ZQ_max+0.01,0.06),
                 pad = 0.15)
cb4.set_label(answer_name,fontsize=10)
cb4.ax.tick_params(labelsize='small',direction='in')


ax4.set(xlabel = 'L(mm)',ylabel = r'$L_{out}$/$L_{in}}$',zlabel='$i_{appl}$ (mA/$cm^2$)')
ax4.xaxis.set_ticks(np.arange(L1, L3+0.01,2.5))
ax4.yaxis.set_ticks(np.arange(bB1, bB3+0.01,0.5))
ax4.zaxis.set_ticks(np.arange(i1,i3+10,60))
ax4.set_zlim(i1,i3)
plt.savefig('L_bB_i_VE.tiff',format='tiff',dpi=600)
```

Algorithm to plot Figure 39

The following algorithm (in python) was developed to plot **Figure 39**

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import numpy as np
import matplotlib as mpl
from matplotlib import cm

# reading data
df1 = pd.read_csv('ohm.txt',sep='\t')
df2 = pd.read_csv('etac.txt',sep='\t')

# selecting matrix X and vector y
X1 = df1.iloc[:, 0:5].values
y1 = df1.iloc[:, 5].values

X2 = df2.iloc[:, 0:5].values
y2 = df2.iloc[:, 5].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1p = poly.fit_transform(X1)
X2p = poly.fit_transform(X2)
regressor = LinearRegression()
regressor.fit(X1p,y1)

# calculating predict values and coefficients
y_prev = regressor.predict(X1p)
coef = regressor.coef_


# preparing data for 4D plots
levels = 300
points = 200
transp = 0.9

i1 = 40
i2 = 100
i3 = 160
bB1 = 0.5
bB2 = 1
bB3 = 1.5
Q1 = 0.15
Q2 = 0.5
Q3 = 0.85
L1 = 2.5
L2 = 5
L3 = 7.5
H1 = 14.5
H2 = 30
H3 = 45.5

L = np.linspace(L1,L3,points)
```

```
bB = np.linspace(bB1,bB3,points)

w = regressor.intercept_
w0,w1,w2,w3,w4,w5 = coef[0],coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]


const7 =
w+w0+w2*i1+w3*Q2+w4*H2+w22*(i1**2)+w23*i1*Q2+w24*i1*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)
const8 =
w+w0+w2*i2+w3*Q2+w4*H2+w22*(i2**2)+w23*i2*Q2+w24*i2*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)
const9 =
w+w0+w2*i3+w3*Q2+w4*H2+w22*(i3**2)+w23*i3*Q2+w24*i3*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)


L = np.linspace(L1,L3,points)
bB = np.linspace(bB1,bB3,points)

L, bB   = np.meshgrid(L, bB)
Z7 =
const7+w1*L+w5*bB+w11*(L**2)+w12*L*i1+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i1*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)
Z8 =
const8+w1*L+w5*bB+w11*(L**2)+w12*L*i2+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i2*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)
Z9 =
const9+w1*L+w5*bB+w11*(L**2)+w12*L*i3+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i3*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)

fig1, axs1 = plt.subplots()

norm1 = mpl.colors.Normalize(vmin=80,vmax=230)
img1=axs1.contourf(L, bB, Z9, levels = levels,alpha=transp)
axs1.set(xlabel = 'L(mm)',ylabel = r'$L_{out}$/$L$')
axs1.xaxis.set_ticks(np.arange(L1, L3+0.01,2.5))
axs1.yaxis.set_ticks(np.arange(bB1, bB3+0.01,0.5))
img1.set_clim(80,230)


# setting regression2
regressor.fit(X2p,y2)
y_prev = regressor.predict(X2p)
coef = regressor.coef_

w = regressor.intercept_
w0,w1,w2,w3,w4,w5 = coef[0],coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]

const7 =
w+w0+w2*i1+w3*Q2+w4*H2+w22*(i1**2)+w23*i1*Q2+w24*i1*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)
const8 =
w+w0+w2*i2+w3*Q2+w4*H2+w22*(i2**2)+w23*i2*Q2+w24*i2*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)
const9 =
w+w0+w2*i3+w3*Q2+w4*H2+w22*(i3**2)+w23*i3*Q2+w24*i3*H2+w33*(Q2**2)+w34*Q2*H2+w44*(H2**2)

Z7 =
const7+w1*L+w5*bB+w11*(L**2)+w12*L*i1+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i1*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)
```

```
Z8 =
const8+w1*L+w5*bB+w11*(L**2)+w12*L*i2+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i2*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)
Z9 =
const9+w1*L+w5*bB+w11*(L**2)+w12*L*i3+w13*L*Q2+w14*L*H2+w15*L*bB+w25*i3*bB+w35*Q2*bB+w4
5*H2*bB+w55*(bB**2)


img2=axs1.contourf(L, bB, Z9, levels = levels,alpha=transp)
axs1.set(xlabel = 'L(mm)')
axs1.xaxis.set_ticks(np.arange(L1, L3+0.01,2.5))
axs1.yaxis.set_ticks(np.arange(bB1, bB3+0.01,0.5))
img2.set_clim(80,230)

norm = mpl.colors.Normalize(vmin=80,vmax=230)
cb4=plt.colorbar(cm.ScalarMappable(norm=norm),
             orientation='vertical',
             ticks=np.arange(80,230+0.01,40))
cb4.set_label('Overpotential (mV)',fontsize=10)
cb4.ax.tick_params(labelsize='small',direction='in')
```

Algorithm to plot Figure 40

The following algorithm (in python) was developed to calculate the maximum voltage efficiency for each applied current density and plot **Figure 40**.

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

```
import sympy as sp
import pandas as pd
import  numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import matplotlib as mpl
from matplotlib import cm


# reading data
file_name='VE'
df_prev = 'inputname.txt'.replace('inputname',file_name)
df = pd.read_csv(df_prev,sep='\t')

# selecting matrix X and vector y
X = df.iloc[:, 0:5].values
y = df.iloc[:, 5].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
regressor = LinearRegression()
regressor.fit(X1,y)

# calculating coefficients
y_prev = regressor.predict(X1)
coef = regressor.coef_
w0,w1,w2,w3,w4,w5 = regressor.intercept_,coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]

# building equations
Q = 0.5
H = 30
Q_array=np.linspace(Q,Q,10)
H_array=np.linspace(H,H,10)
I = np.arange(25,161,15)
L,bB = sp.symbols(['L','bB'])
L_plot=[]
bB_plot=[]

for i in range(len(I)):
    const =
w0+w2*I[i]+w3*Q+w4*H+w22*(I[i]**2)+w23*I[i]*Q+w24*I[i]*H+w33*(Q**2)+w34*Q*H+w44*(H**2)
    eq =
const+w1*L+w5*bB+w11*(L**2)+w12*L*I[i]+w13*L*Q+w14*L*H+w15*L*bB+w25*I[i]*bB+w35*Q*bB+w45*
H*bB+w55*(bB**2)
```

```
    # differentiation
    deq_dL = sp.diff(eq,L)
    deq_dbB = sp.diff(eq,bB)

    #system equations
    mat_A = sp.linear_eq_to_matrix([deq_dL,deq_dbB],[L,bB])
    mat_X = sp.Matrix([[L,bB]]).T
    system = mat_A[0]*mat_X-mat_A[1]
    ans = sp.solve(system)
    a = round(float(ans[L]),2)
    b = round(float(ans[bB]),2)
    L_plot.append(a)
    bB_plot.append(b)


# testing
df_pred=pd.DataFrame({'L_e':L_plot,'i_appl':I,'Q':Q_array,'H':H_array,
                      'b/B':bB_plot})
X2 = df_pred.iloc[:,:].values
X3=poly.fit_transform(X2)
VE_max=regressor.predict(X3)


# plots
plt.figure(figsize=[4.8,4.8])
c=np.linspace(25,175,10)
plt.subplot(211)
plt.scatter(L_plot,bB_plot,c=c)
plt.xlabel('L (mm)',fontsize=10)
plt.ylabel( r'$L_{out}$/$L_{in}$',fontsize=10,labelpad=1.1)
plt.xticks(np.arange(2.5,7.7+0.01,1.5),fontsize=9)
plt.xlim(2.5,7.7)
plt.yticks(np.arange(0.5,1.3+0.01,0.4),fontsize=9)
plt.ylim(0.5,1.3)
norm = mpl.colors.Normalize(vmin=I.min(),vmax=I.max())
cb = plt.colorbar(cm.ScalarMappable(norm=norm),
             orientation='vertical',
             ticks=np.arange(I.min(),I.max()+0.01,40),
             pad = 0.04,shrink=1.0)
cb.set_label('$i_{appl}$ (mA/$cm^2$)',fontsize=10)
cb.ax.tick_params(labelsize='small',direction='in')

plt.subplot(212)
plt.scatter(I,VE_max,color='black')
plt.xlabel('$i_{appl}$  (mA/$cm^2$)',fontsize=10)
plt.ylabel( 'Maximum VE (%)',fontsize=10,labelpad=1)
plt.yticks(np.arange(50,100+0.1,10),fontsize=9)
plt.xticks(np.arange(25,165,40),fontsize=9)
plt.ylim(50,100)
plt.xlim(20,165,40)

plt.subplots_adjust(hspace=0.4)

plt.savefig('name.tiff',format='tiff',dpi=600)
```

Algorithm to calculate percentage of mitigation (Equation 145).

The following algorithm (in python) was developed to calculate the percentage of mitigation.

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

```
import pandas as pd
import  numpy as np
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression



file_name=str(input('File name: '))
df_prev = 'inputname.txt'.replace('inputname',file_name)
df = pd.read_csv(df_prev,sep='\t')

# selecting matrix X and vector y
X = df.iloc[:, 0:5].values
y = df.iloc[:, 5].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
regressor = LinearRegression()
regressor.fit(X1,y)

Xt = poly.fit_transform(X_t)
pred = regressor.predict(Xt)

#deffing values
Q = 0.5
H = 30

bB1 = 0.5
bB3 = 1.5
L1 = 2.5
L3 = 7.5

points = 200
I = np.linspace(25,160,25)
L = np.linspace(L1,L3,points)
bB = np.linspace(bB1,bB3,points)
points = 200
L, bB   = np.meshgrid(L, bB)

# calculating coefficients
y_prev = regressor.predict(X1)
coef = regressor.coef_
coef[0] = regressor.intercept_
w0,w1,w2,w3,w4,w5 = coef[0],coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]
```

```
# determination of mitigation
eta_max=[]

for i in range(len(I)):
    const =
w0+w2*I[i]+w3*Q+w4*H+w22*(I[i]**2)+w23*I[i]*Q+w24*I[i]*H+w33*(Q**2)+w34*Q*H+w44*(H**2)
    eq =
const+w1*L+w5*bB+w11*(L**2)+w12*L*I[i]+w13*L*Q+w14*L*H+w15*L*bB+w25*I[i]*bB+w35*Q*bB+w45*
H*bB+w55*(bB**2)
    i_max = eq.max()
    eta_max.append(i_max)


eta100 = 100*pred
etap = eta100/eta_max
etam = 100-etap
```

Algorithm to calculate the optimal geometry (Equation 146, Equation 147 and Figure 43**).**

The following algorithm (in python) was developed to calculate the general optimal geometry and plot **Figure 43**.

Input files: data from **Table 12**. Doehlert design for five variables of Study 1 – codified and **Table 22.** All responses from Doehlert design (Study 1)*.

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp

n_points=13
# reading data
file_name='VE'
df_prev = 'inputname.txt'.replace('inputname',file_name)
df = pd.read_csv(df_prev,sep='\t')



# selecting matrix X and vector y
X = df.iloc[:, 0:5].values
y = df.iloc[:, 5].values


# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
regressor = LinearRegression()
regressor.fit(X1,y)

# calculating predict values and coefficients
y_prev = regressor.predict(X1)
coef = regressor.coef_
inpt = regressor.intercept_


# maximum efficiency

w0,w1,w2,w3,w4,w5 = regressor.intercept_,coef[1],coef[2],coef[3],coef[4],coef[5]
w11,w12,w13,w14,w15 = coef[6],coef[7],coef[8],coef[9],coef[10]
w22,w23,w24,w25 = coef[11],coef[12],coef[13],coef[14]
w33,w34,w35 = coef[15],coef[16],coef[17]
w44,w45 = coef[18],coef[19]
w55 = coef[20]


Q = 0.5
H = 30
Q_array=np.linspace(Q,Q,n_points)
H_array=np.linspace(H,H,n_points)
I = np.linspace(25,161,n_points)
L,bB = sp.symbols(['L','bB'])
```

```
L_plot=[]
bB_plot=[]

for i in range(len(I)):
    const =
w0+w2*I[i]+w3*Q+w4*H+w22*(I[i]**2)+w23*I[i]*Q+w24*I[i]*H+w33*(Q**2)+w34*Q*H+w44*(H**2)
    eq =
const+w1*L+w5*bB+w11*(L**2)+w12*L*I[i]+w13*L*Q+w14*L*H+w15*L*bB+w25*I[i]*bB+w35*Q*bB+w45*
H*bB+w55*(bB**2)

    # differentiation
    deq_dL = sp.diff(eq,L)
    deq_dbB = sp.diff(eq,bB)

    #system equations
    mat_A = sp.linear_eq_to_matrix([deq_dL,deq_dbB],[L,bB])
    mat_X = sp.Matrix([[L,bB]]).T
    system = mat_A[0]*mat_X-mat_A[1]
    ans = sp.solve(system)
    a = round(float(ans[L]),2)
    b = round(float(ans[bB]),2)
    L_plot.append(a)
    bB_plot.append(b)



df_pred=pd.DataFrame({'L_e':L_plot,'i_appl':I,'Q':Q_array,'H':H_array,
                      'b/B':bB_plot})
X2 = df_pred.iloc[:,:].values
X3=poly.fit_transform(X2)
VE_max=regressor.predict(X3)


### finding other values from VE
VE_values=[]
for i in range(len(L_plot)):

    df_pred=pd.DataFrame({'L_e':np.linspace(L_plot[i],L_plot[i],n_points),'i_appl':I,'Q':Q_array,'H':H_array,
                      'b/B':np.linspace(bB_plot[i],bB_plot[i],n_points)})
    X2 = df_pred.iloc[:,:].values
    X3=poly.fit_transform(X2)
    y_prev2=regressor.predict(X3)
    VE_values.append(y_prev2)


# finding best design

var = []
for i in range(len(I)):
    dif = VE_values[i]-VE_max
    var.append(dif.max()-dif.min())


#plotting
fig, (ax1,ax2) = plt.subplots(1,2)

ax2.set_title('(b)')
ax2.plot(I,var,'k--')
ax2.scatter(I[6],var[6],label='Optimal geometry',color='red')
ax2.set_xlabel('Geometric condition')
ax2.tick_params(axis='x',bottom=False,labelbottom=False)
ax2.set_ylabel('$\delta$ (%)')
ax2.set_yticks(np.arange(0,3,1))
ax2.legend()
```

```
ax1.set_title('(a)')
ax1.plot(VE_max*-1+VE_values[0],color='yellow',label='25 mA/$cm^2$')
ax1.plot(VE_max*-1+VE_values[3],color='green',label='59 mA/$cm^2$')
ax1.plot(VE_max*-1+VE_values[9],color='blue',label='127 mA/$cm^2$')
ax1.plot(VE_max*-1+VE_values[12],color='indigo',label='161 mA/$cm^2$')
ax1.legend()
ax1.set_xlabel('Geometric condition')
ax1.tick_params(axis='x',bottom=False,labelbottom=False)
ax1.set_ylabel('$\Delta$VE (%)')
ax1.set_yticks(np.arange(0,-1.51,-0.5))

fig.tight_layout(pad=2.0)

plt.savefig('final_strategy.tiff',format='tiff',dpi=300)
```

Algorithm for validation of regression analysis (Figure 44 **and Table** 26**).**

The following algorithm (in python) was developed to validate the regression analysis and plot the **Figure 44**.

Input files: validation data.

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,r2_score
import matplotlib.pyplot as plt

# loading files
file_training=str(input('File training input: '))
df_prev = 'inputname.txt'.replace('inputname',file_training)
df = pd.read_csv(df_prev,sep='\t')

file_test=str(input('File test input:'))
df_prev2 = 'inputname.txt'.replace('inputname',file_test)
df2 = pd.read_csv(df_prev2,sep='\t')

# selecting matrix X and vector y
X_train = df.iloc[:, 0:5].values
y_train = df.iloc[:, 5].values
X_test = df2.iloc[:,0:5].values
y_test = df2.iloc[:,5].values

# setting regression
poly = PolynomialFeatures(degree = 2)

X1 = poly.fit_transform(X_train)
X2 = poly.fit_transform(X_test)

regressor = LinearRegression()
training = regressor.fit(X1,y_train)


#prediction
y_prev = regressor.predict(X2)

#metrics
RMSE = mean_squared_error(y_test, y_prev,squared=False)
r2 = r2_score(y_test,y_prev)
r2s = str(round(r2,4))
RMSEs = str(round(RMSE,2))


#plots
answer_name = str(input('Answer name (unit): '))
fig1, ax1 = plt.subplots()
ax1.plot(np.linspace(round(y_test.min()),round(y_test.max()),200),np.linspace(round(y_test.min()),round(y_test.max(
)),200),'k--')
ax1.plot(y_test,y_prev,'bo',markersize=7)
ax1.set(xlabel = 'Simulated '+ answer_name,ylabel = 'Predicted '+ answer_name)
plt.grid()
plt.text(140,20,' $R^2$:' + r2s + '\n RMSE:' + RMSEs)
plt.savefig('Regression_'+file_test+'.tiff',format='tiff',dpi=600)
```

Appendix B – Scripts for data treatment (Study 2)

Algorithm to determine electrolyte viscosity (Figure 46)

The following algorithm (in python) was developed to determine the positive and negative electrolyte viscosity.

Input files: data from [72,73].

For regression:

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,r2_score
import matplotlib.pyplot as plt
import numpy as np



# importing data
base = pd.read_csv('file_name.txt',sep='\t')
X = base.iloc[:, 0:4].values
y = base.iloc[:, 4].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)

regressor = LinearRegression()


regressor.fit(X1, y)
y_prev = regressor.predict(X1)

r2 = r2_score(y,y_prev)
mse = mean_absolute_error(y, y_prev)

SoC = np.linspace(0.1,0.9,20)
c_act = 1.48
c_sup = 3.0
T = 298.15

mi_prev = []
for i in range(len(SoC)):
    prev = np.array([SoC[i],c_act,c_sup,T]).reshape(1,-1)
    prev_p = poly.fit_transform(prev)
    pred=float(regressor.predict(prev_p))
    mi_prev.append(pred)

std = round(np.array(mi_prev).std(),2)
mean = round(np.array(mi_prev).mean(),2)

mean_str = str(mean)
std_str = str(std)
```

```
result = mean_str+' +/- '+std_str
print(result)

plt.plot(np.linspace(2,10,200),np.linspace(2,10,200),'k--')
plt.plot(y,y_prev,'bo',markersize=7)
plt.xlabel('Experimental Viscosity (mPa.s)')
plt.ylabel( 'Predicted Viscosity (mPa.s)')

plt.savefig('figure_name.tiff',formate='tiff')
```

For cross-validation:

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import numpy as np
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_validate
from sklearn.model_selection import KFold

# importing data
base = pd.read_csv('file_name.txt',sep='\t')

X = base.iloc[:, 0:4].values
y = base.iloc[:, 4].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X_poly = poly.fit_transform(X)
regressor = LinearRegression()

# cross-validation
r2_list = []
mae_list = []

for i in range(30):

    kf=KFold(n_splits=5, random_state=i, shuffle=True)

    cv_results = cross_validate(regressor,X_poly,y,cv=kf,return_estimator=True)

    a=cross_val_score(regressor, X_poly, y,cv=kf,scoring='r2').mean()
    b=np.sqrt(abs(cross_val_score(regressor, X_poly, y,scoring='neg_mean_absolute_error',cv=kf))).mean()
    r2_list.append(a)
    mae_list.append(b)

r2 = np.array(r2_list).mean()
mae = np.array(mae_list).mean()
```

Algorithm to calculate the responses from COMSOL files (Table 28 **and Table** 29**)**

The following algorithm (in python) was developed to calculate several responses from the designed experiments: capacity loss rate in the first 20th cycles (CL_pc_mean_20), the capacity loss rate of all cycles (CL_pc_mean), capacity loss at 20th cycle (C_20), capacity loss at final cycle (C_final), mean voltage efficiency (VE_mean), mean coulombic efficiency (CE_mean) and mean energy efficiency (EE_mean).

Input files: (COMSOL files) cell potential versus time, current versus time.

```
import pandas as pd
import numpy as np
from scipy.signal import find_peaks
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
from scipy.integrate import trapz

# reading files
Vt = input('Name of file V versus time:')
it = input('Name of file i versus time:')
#code = str(input('Code:'))

file_Vt = 'inputname.txt'.replace('inputname',Vt)
file_it = 'inputname.txt'.replace('inputname',it)

df1_raw = pd.read_csv(file_Vt,skiprows=(7))
df2_raw = pd.read_csv(file_it,skiprows=(7))

df1 = df1_raw.rename(columns={'% X':'time','Height':'V'})
df2 = df2_raw.rename(columns={'% X':'time','Height':'i'})


# selecting data
x = list(df1['time'])
y = list(df1['V'])

x2 = list(df2['time'])
y2 = list(df2['i'])


#interpolation
it=200000
f= interp1d(x,y)
x_V= np.linspace(np.array(x).min(),np.array(x).max(),it)
y_V = f(x_V)


f2= interp1d(x2,y2)
x_i= np.linspace(np.array(x2).min(),np.array(x2).max(),it)
```

```
y_i = f2(x_i)


# differentiation of discrete values of V versus time data (for time peak determination)
dydx_V= np.gradient(y_V,x_V)

# finding peaks charge/discharge
peaks = find_peaks(abs(dydx_V),height=0.08)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
   height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x_V)[peaks[0]]
if peak_pos.shape[0] == 1:
   peak_pos = np.append(np.array(x_V)[peaks[0]],x_V[len(x_V)-1])


# plotting peaks
fig = plt.figure()
ax = fig.subplots()
ax.plot(x_V,abs(dydx_V),'-k', label = 'Time derivative')
ax.scatter(peak_pos, height, color = 'r', s = 15, marker = 'D', label = 'Peaks')
ax.legend()
ax.grid()


#calculating capacity loss
step_time = []
step_time.append(peak_pos[0])
z_ite = 0
for i in range(len(peak_pos)):
   if z_ite == len(peak_pos)-1:
      break
   step_time.append(peak_pos[z_ite+1]-peak_pos[z_ite])
   z_ite += 1
discharge_time = step_time[1:len(step_time):2]
capacity_loss = []
for i in range(len(discharge_time)):
   capacity_loss.append(100*discharge_time[i]/discharge_time[0])

n_cycles = np.arange(1,len(capacity_loss)+1,1)
fig1,ax1 = plt.subplots()
ax1.scatter(n_cycles,capacity_loss,c='black')
ax1.set(xlabel = 'Cycle number',ylabel = 'Discharge capacity(%)')
#plt.savefig('capacity_loss_'+code+'.tiff',format='tiff',dpi=600)



# calculating capacity loss per cycle
CL_pc_list = []

for i in range(len(capacity_loss)-1):
   CL_pc = capacity_loss[i]-capacity_loss[i+1]
   CL_pc_list.append(CL_pc)

CL_pc_mean_20 = np.array(CL_pc_list[0:20]).mean()
CL_pc_mean_total = np.array(CL_pc_list).mean()
C_20 = abs(100-capacity_loss[19])
C_final = abs(100-capacity_loss[len(capacity_loss)-1])


# preparing index, dataframes arrays for efficiencies calculation
discharge_steps = peak_pos[1:len(peak_pos):2]
charge_steps = peak_pos[0:len(peak_pos):2]


charge_time = step_time[0:len(step_time):2]
```

```
a = 0
index_discharge = []
while a < discharge_steps.shape[0]:
    for i in range(x_V.shape[0]):
        if x_V[i] == discharge_steps[a]:
            index_discharge.append(i)
    a += 1


b = 0
index_charge = []
while b < (charge_steps.shape[0]):
    for i in range(x_V.shape[0]):
        if x_V[i] == charge_steps[b]:
            index_charge.append(i)
    b += 1


# calculating voltage efficiency
Avg_V = []
Avg_V.append((trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))/charge_time[0])

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    Avg_dis =
(trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))/discharge_time[e]
    Avg_V.append(Avg_dis)
    if x_ite == len(charge_time):
      break
    Avg_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))/charge_time[
x_ite]
    Avg_V.append(Avg_cha)
    e += 1
    x_ite +=1

VE_prev = []

for i in range(len(Avg_V)-1):
    h = 100*Avg_V[i+1]/Avg_V[i]
    VE_prev.append(h)

VE = VE_prev[0::2]

VE_std = np.array(VE).std()
VE_mean = np.array(VE).mean()

for i in range(len(VE)):
    if VE[i] > VE_mean + VE_std:
        VE[i] = VE_mean
    elif VE[i] < VE_mean - VE_std:
        VE[i] = VE_mean

#calculating coulombic efficiency based on time
CE = []
q_ite = 0

for i in range(n+1):
    r_const = discharge_time[q_ite]*100/charge_time[q_ite]
    CE.append(r_const)
    q_ite += 1
```

```
#calculating energy efficiency
EE = []
s_ite = 0
for i in range(len(VE)):
    t_const = VE[s_ite]*CE[s_ite]/100
    EE.append(t_const)
    s_ite += 1

#creating and exporting DataFrames
n_cycles = np.arange(1,n+2,1)

CE_mean = (np.array(CE)[0:20]).mean()
VE_mean = (np.array(VE)[0:20]).mean()
EE_mean = (np.array(EE)[0:20]).mean()

mean_values = pd.DataFrame(['CL_rat and mean values',np.round(CL_pc_mean_20,4),
                            np.round(CL_pc_mean_total,4),
                            np.round(C_20,2),
                            np.round(C_final,2),
                            np.round(VE_mean,2),
                            np.round(CE_mean,2),
                            np.round(EE_mean,2)]).T



#plotting graphs
fig2,ax2 = plt.subplots()
ax2.plot(n_cycles,np.array(VE),'bo',label = 'Voltage')
ax2.plot(n_cycles,np.array(CE),'ro',label = 'Coulombic')
ax2.plot(n_cycles,np.array(EE),'go',label = 'Energy')
ax2.set(xlabel = 'Cycle number',ylabel = 'Efficiency (%)', title = 'Efficiencies')
plt.ylim(50,100)
plt.legend()
plt.savefig('efficiencies_'+code+'.tiff',format='tiff',dpi=600)


print(mean_values)
```

Algorithm to calculate effects (Figure 49**)**

The following algorithm (in python) was developed to calculate the effects of 2^3 factorial design

Input files: data from **Table 17** and **Table 28**.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

df=pd.read_csv('file_name.txt',sep='\t')

X = df.iloc[:,0:3].values
y = df.iloc[:,3].values

effects = []

# primary effects
list1 = []
list2 = []
list3 = []


for i in range(len(y)):
   it1 = X[i][0]*y[i]
   list1.append(it1)
   it2 = X[i][1]*y[i]
   list2.append(it2)
   it3 = X[i][2]*y[i]
   list3.append(it3)

ef1=2*np.array(list1).mean()
effects.append(ef1)
ef2=2*np.array(list2).mean()
effects.append(ef2)
ef3=2*np.array(list3).mean()
effects.append(ef3)



#secondary effects
list12 = []
list13 = []
list23 = []



for i in range(len(y)):
   it12 = X[i][0]*X[i][1]*y[i]
   list12.append(it12)
   it13 = X[i][0]*X[i][2]*y[i]
   list13.append(it13)

   it23 = X[i][1]*X[i][2]*y[i]
```

```
    list23.append(it23)


ef12=2*np.array(list12).mean()
effects.append(ef12)
ef13=2*np.array(list13).mean()
effects.append(ef13)


ef23=2*np.array(list23).mean()
effects.append(ef23)



#terciary effects
list123 = []


for i in range(len(y)):
    it123 = X[i][0]*X[i][1]*X[i][2]*y[i]
    list123.append(it123)


ef123=2*np.array(list123).mean()
effects.append(ef123)


#creating plots
effects_name=['1','2','3','12','13','23','123']


df_effects = pd.DataFrame({'effect':effects_name,'Value':effects})

df_effects_or = df_effects.sort_values(by='Value',ignore_index=True)

#porcentage graph
effects_squared = []

for i in range(len(effects)):
    effects_squared.append(abs(effects[i]))

sum_effects_squared = np.array(effects_squared).sum()

percentage_ef = []
for i in range(len(effects_squared)):
    b_ite = 100*effects_squared[i]/sum_effects_squared
    percentage_ef.append(b_ite)

df_perc = pd.DataFrame({'effect':effects_name,'percentage':percentage_ef})
df_perc_or = df_perc.sort_values(by='percentage',ascending=False,ignore_index=True)
percentage_ef = df_perc_or.iloc[:,1].values
name_ef_perc = df_perc_or.iloc[:,0]

fig,(ax,ax3) = plt.subplots(1,2)
fig.set_figwidth(10)
ax.bar(name_ef_perc,percentage_ef,color='green')
ax.set_xlabel('Standardized Effect')
ax.set_ylabel('Percentage (%)')
ax.yaxis.label.set_color('green')
ax.tick_params(axis='y', colors='green')

perc_cum = percentage_ef.cumsum()

ax2 = ax.twinx()
ax2.plot(name_ef_perc,perc_cum,marker='o',color='black')
```

```
ax2.set_ylabel('Cumulative Sum (%)')
ax.set_title('(a)')


#probability graph and pareto
effect_or = df_effects_or.iloc[:,1]
name_or = df_effects_or.iloc[:,0]

initial = []
initial.append(0)
for i in range(len(effects)-1):
    c_ite = 1/len(effects)+initial[i]
    initial.append(c_ite)

final = []
for i in range(len(initial)):
    final.append(initial[i+1])
    if len(initial) == len(final)+1:
        break
final.append(1)

middle = []
for i in range(len(initial)):
    d_ite = (initial[i]+final[i])/2
    middle.append(d_ite)

inv_norm = norm.ppf(middle)


ax3.scatter(effect_or,inv_norm,color='green')
ax3.set_xlabel('Effect on $CL_{rate}$ (%)')
ax3.set_ylabel('z value')
yline = np.linspace(np.array(inv_norm).min(),np.array(inv_norm).max(),10)
xline = np.linspace(0,0,10)
ax3.plot(xline,yline,color='black')
ax3.set_title('(b)')

plt.subplots_adjust(wspace=0.5)

for i, txt in enumerate(name_or):
    ax3.annotate(txt,(effect_or[i],inv_norm[i]))

plt.text(11,45,'$i_{appl}$')
plt.text(16.7,100,'$c_{act}$')
plt.savefig('Fig_name.tiff',format='tiff',dpi=300)
```

Algorithm to calculate integrated net fluxes (Figure 50 **and Figure** 53**)**

The following algorithm (in python) was developed to calculate the integrated net fluxes across the membrane: total flux (tflux), diffusional flux (dflux), migration flux (mflux) and convective flux (cflux).

Input files: (COMSOL files) total, diffusional, migrational and convective flux across membrane for V(II), V(III), V(IV) and V(V).

```
import pandas as pd
import numpy as np
from scipy.signal import find_peaks
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
from scipy.integrate import trapz

# reading files
df = pd.read_csv('vanadium_species_file.txt',skiprows=7)
df = df.rename(columns={'% X':'time','Height':'V'})

df1 = pd.read_csv('flux2_V2.txt',skiprows=7)
df2 = pd.read_csv('flux2_V3.txt',skiprows=7)
df3 = pd.read_csv('flux2_V4.txt',skiprows=7)
df4 = pd.read_csv('flux2_V5.txt',skiprows=7)

df1 = df1.rename(columns={'% X':'time','Height':'tflux','Height.1':'dflux','Height.2':'mflux','Height.3':'cflux'})
df2 = df2.rename(columns={'% X':'time','Height':'tflux','Height.1':'dflux','Height.2':'mflux','Height.3':'cflux'})
df3 = df3.rename(columns={'% X':'time','Height':'tflux','Height.1':'dflux','Height.2':'mflux','Height.3':'cflux'})
df4 = df4.rename(columns={'% X':'time','Height':'tflux','Height.1':'dflux','Height.2':'mflux','Height.3':'cflux'})



# selecting data
x = list(df['time'])
y = list(df['V'])

ft=1e3

total1 = df1['tflux'].values*ft
dif1 = df1['dflux'].values*ft
mig1 = df1['mflux'].values*ft
conv1 = df1['cflux'].values*ft

total2 = df2['tflux'].values*ft
dif2 = df2['dflux'].values*ft
mig2 = df2['mflux'].values*ft
conv2 = df2['cflux'].values*ft

total3 = df3['tflux'].values*ft
dif3 = df3['dflux'].values*ft
mig3 = df3['mflux'].values*ft
conv3 = df3['cflux'].values*ft

total4 = df4['tflux'].values*ft
dif4 = df4['dflux'].values*ft
mig4 = df4['mflux'].values*ft
```

```
conv4 = df4['cflux'].values*ft

t = df1['time'].values


#interpolation
it=200000
f= interp1d(x,y)
x_V= np.linspace(np.array(x).min(),np.array(x).max(),it)
y_V = f(x_V)



# differentiation of discrete values of V versus time data (for time peak determination)
dydx_V= np.gradient(y_V,x_V)

# finding peaks charge/discharge
peaks = find_peaks(abs(dydx_V),height=0.00002,threshold=0.00002)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
    height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x_V)[peaks[0]]
if peak_pos.shape[0] == 1:
    peak_pos = np.append(np.array(x_V)[peaks[0]],x_V[len(x_V)-1])


#calculating discharge/charge indexes and times
step_time = []
step_time.append(peak_pos[0])
z_ite = 0
for i in range(len(peak_pos)):
    if z_ite == len(peak_pos)-1:
        break
    step_time.append(peak_pos[z_ite+1]-peak_pos[z_ite])
    z_ite += 1
discharge_time = step_time[1:len(step_time):2]


discharge_steps = peak_pos[1:len(peak_pos):2]
charge_steps = peak_pos[0:len(peak_pos):2]


charge_time = step_time[0:len(step_time):2]

a = 0
index_discharge = []
while a < discharge_steps.shape[0]:
    for i in range(x_V.shape[0]):
        if x_V[i] == discharge_steps[a]:
            index_discharge.append(i)
    a += 1


b = 0
index_charge = []
while b < (charge_steps.shape[0]):
    for i in range(x_V.shape[0]):
        if x_V[i] == charge_steps[b]:
            index_charge.append(i)
    b += 1

n_cycles = np.arange(1,len(height)+1,1)


#integrating fluxes V2
```

```
f= interp1d(t,total1)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

itotal_cha1 = np.array(iflux[0:len(iflux):2]).mean()
itotal_dis1 = np.array(iflux[1:len(iflux):2]).mean()

itotal1 = np.array(iflux).mean()


f= interp1d(t,dif1)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

idif_cha1 = np.array(iflux[0:len(iflux):2]).mean()
idif_dis1 = np.array(iflux[1:len(iflux):2]).mean()

idif1 = np.array(iflux).mean()



f= interp1d(t,mig1)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
```

```
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

imig_cha1 = np.array(iflux[0:len(iflux):2]).mean()
imig_dis1 = np.array(iflux[1:len(iflux):2]).mean()

imig1 = np.array(iflux).mean()



f= interp1d(t,conv1)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

iconv_cha1 = np.array(iflux[0:len(iflux):2]).mean()
iconv_dis1 = np.array(iflux[1:len(iflux):2]).mean()


iconv1 = np.array(iflux).mean()



#integrating fluxes V3

f= interp1d(t,total2)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
```

```
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

itotal_cha2 = np.array(iflux[0:len(iflux):2]).mean()
itotal_dis2 = np.array(iflux[1:len(iflux):2]).mean()

itotal2 = np.array(iflux).mean()


f= interp1d(t,dif2)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

idif_cha2 = np.array(iflux[0:len(iflux):2]).mean()
idif_dis2 = np.array(iflux[1:len(iflux):2]).mean()

idif2 = np.array(iflux).mean()



f= interp1d(t,mig2)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
```

```
    e += 1
    x_ite +=1

imig_cha2 = np.array(iflux[0:len(iflux):2]).mean()
imig_dis2 = np.array(iflux[1:len(iflux):2]).mean()

imig2 = np.array(iflux).mean()



f= interp1d(t,conv2)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

iconv_cha2 = np.array(iflux[0:len(iflux):2]).mean()
iconv_dis2 = np.array(iflux[1:len(iflux):2]).mean()


iconv2 = np.array(iflux).mean()


#integrating fluxes V4

f= interp1d(t,total3)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

itotal_cha3 = np.array(iflux[0:len(iflux):2]).mean()
itotal_dis3 = np.array(iflux[1:len(iflux):2]).mean()
```

```
itotal3 = np.array(iflux).mean()


f= interp1d(t,dif3)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

idif_cha3 = np.array(iflux[0:len(iflux):2]).mean()
idif_dis3 = np.array(iflux[1:len(iflux):2]).mean()

idif3 = np.array(iflux).mean()



f= interp1d(t,mig3)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

imig_cha3 = np.array(iflux[0:len(iflux):2]).mean()
imig_dis3 = np.array(iflux[1:len(iflux):2]).mean()

imig3 = np.array(iflux).mean()



f= interp1d(t,conv3)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)
```

```
iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

iconv_cha3 = np.array(iflux[0:len(iflux):2]).mean()
iconv_dis3 = np.array(iflux[1:len(iflux):2]).mean()


iconv3 = np.array(iflux).mean()


#integrating fluxes V5

f= interp1d(t,total4)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
       break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

itotal_cha4 = np.array(iflux[0:len(iflux):2]).mean()
itotal_dis4 = np.array(iflux[1:len(iflux):2]).mean()

itotal4 = np.array(iflux).mean()


f= interp1d(t,dif4)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
```

```
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

idif_cha4 = np.array(iflux[0:len(iflux):2]).mean()
idif_dis4 = np.array(iflux[1:len(iflux):2]).mean()

idif4 = np.array(iflux).mean()



f= interp1d(t,mig4)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

imig_cha4 = np.array(iflux[0:len(iflux):2]).mean()
imig_dis4 = np.array(iflux[1:len(iflux):2]).mean()

imig4 = np.array(iflux).mean()



f= interp1d(t,conv4)
x_V= np.linspace(np.array(t).min(),np.array(t).max(),it)
y_V = f(x_V)

iflux= []
iflux.append(trapz(y_V[0:index_charge[0]],x=x_V[0:index_charge[0]]))

e = 0
x_ite = 1
n = int((discharge_steps.shape[0]+charge_steps.shape[0])/2-1)
for i in range(n+1):
    if e == len(discharge_time):
        break
    iflux_dis = (trapz(y_V[index_charge[e]:index_discharge[e]],x=x_V[index_charge[e]:index_discharge[e]]))
    iflux.append(iflux_dis)
    if x_ite == len(charge_time):
```

```
      break
    iflux_cha =
(trapz(y_V[index_discharge[e]:index_charge[x_ite]],x=x_V[index_discharge[e]:index_charge[x_ite]]))
    iflux.append(iflux_cha)
    e += 1
    x_ite +=1

iconv_cha4 = np.array(iflux[0:len(iflux):2]).mean()
iconv_dis4 = np.array(iflux[1:len(iflux):2]).mean()


iconv4 = np.array(iflux).mean()


# plotting

fig,ax = plt.subplots(nrows=2,ncols=2)
plt.subplot(2,2,1)
plt.bar('total',itotal1)
plt.bar('dif',idif1)
plt.bar('mig',imig1)
plt.bar('conv',iconv1)
plt.ylim([-0.5,3])

plt.subplot(2,2,2)
plt.bar('total',itotal2)
plt.bar('dif',idif2)
plt.bar('mig',imig2)
plt.bar('conv',iconv2)
plt.ylim([-0.5,3])


plt.subplot(2,2,3)
plt.bar('total',itotal3)
plt.bar('dif',idif3)
plt.bar('mig',imig3)
plt.bar('conv',iconv3)
plt.ylim([-3,0.5])

plt.subplot(2,2,4)
plt.bar('total',itotal4)
plt.bar('dif',idif4)
plt.bar('mig',imig4)
plt.bar('conv',iconv4)
plt.ylim([-3,0.5])

plt.savefig('fig_name.tiff',format='tiff',dpi=300)
```

Algorithm to calculate the V(V) limiting concentration (Figure 51 **and Figure** 54**)**

The following algorithm (in python) was developed to calculate the V(V) limiting concentration and its derivative.

Input files: (COMSOL files) V(V) concentration versus time.

```
import pandas as pd
import numpy as np
from scipy.signal import find_peaks
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

df1 = pd.read_csv('c1.txt',skiprows=7)
df2 = pd.read_csv('c2.txt',skiprows=7)
df3 = pd.read_csv('c3.txt',skiprows=7)

df1 = df1.rename(columns={'% X':'time','Height':'cV2','Height.1':'cV3','Height.2':'cV4','Height.3':'cV5'})
df2 = df2.rename(columns={'% X':'time','Height':'cV2','Height.1':'cV3','Height.2':'cV4','Height.3':'cV5'})
df3 = df3.rename(columns={'% X':'time','Height':'cV2','Height.1':'cV3','Height.2':'cV4','Height.3':'cV5'})


cV5_1 = df1['cV5'].values*-1
cV5_2 = df2['cV5'].values*-1
cV5_3 = df3['cV5'].values*-1


t1 = df1['time'].values
t2 = df2['time'].values
t3 = df3['time'].values

it=200000
f= interp1d(t1,cV5_1)
x= np.linspace(np.array(t1).min(),np.array(t1).max(),it)
y = f(x)

peaks = find_peaks(y,height=-200,threshold=0.000000002)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
   height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x)[peaks[0]]
if peak_pos.shape[0] == 1:
   peak_pos = np.append(np.array(x)[peaks[0]],x[len(x)-1])

fig = plt.figure()
ax = fig.subplots()
ax.plot(x,y,'-k', label = 'Time derivative')
ax.scatter(peak_pos, height, color = 'r', s = 15, marker = 'D', label = 'Peaks')
ax.legend()
ax.grid()


n_cycles1 = np.arange(1,len(height)+1,1)
```

```
Max = height*-1*0.001

it2 = 18
f2= interp1d(n_cycles1,Max)
x2= np.linspace(np.array(n_cycles1).min(),np.array(n_cycles1).max(),it2)
y2 = f2(x2)


d_Max1 = np.gradient(y2,x2)*1e3

fig = plt.figure()
ax = fig.subplots()
ax.plot(n_cycles1,Max,'ko')



ax2=ax.twinx()
ax2.plot(x2,d_Max1,'k-',label='1.10 mol/L')



f= interp1d(t3,cV5_3)
x= np.linspace(np.array(t2).min(),np.array(t2).max(),it)
y = f(x)

peaks = find_peaks(y,height=-200,threshold=0.000000002)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
   height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x)[peaks[0]]
if peak_pos.shape[0] == 1:
   peak_pos = np.append(np.array(x)[peaks[0]],x[len(x)-1])

n_cycles2 = np.arange(1,len(height)+1,1)
Max2 = height*-1*.001
ax.plot(n_cycles2,Max2,'bo')


f2= interp1d(n_cycles2,Max2)
x2= np.linspace(np.array(n_cycles2).min(),np.array(n_cycles2).max(),it2)
y2 = f2(x2)


d_Max2 = np.gradient(y2,x2)*1e3
ax2.plot(x2,d_Max2,'b-',label='1.80 mol/L')
ax2.set_yticks(np.arange(-10,0,3))


ax.set_xlabel('Cycle number')
ax.set_ylabel('V(V) limiting concentration (mol/L)')
ax2.set_ylabel('Derivative (X10$^3$ mol/L)')
plt.xlim([0,20.5])
ax.set_xticks(np.arange(0,20.1,5))


"""f= interp1d(t3,cV5_3)
x= np.linspace(np.array(t3).min(),np.array(t3).max(),it)
y = f(x)

peaks = find_peaks(y,height=-200.,threshold=0.000000002)
height = peaks[1]['peak_heights']
if height.shape[0] == 1:
   height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

peak_pos = np.array(x)[peaks[0]]
if peak_pos.shape[0] == 1:
```

```
    peak_pos = np.append(np.array(x)[peaks[0]],x[len(x)-1])

n_cycles2 = np.arange(1,len(height)+1,1)
SoH_max2 = height
ax.plot(n_cycles2,SoH_max2*-1,'g-',label='c2,i1')

plt.legend()
ax.set_xlabel('Cycle number')
ax.set_ylabel('V5 concentration')"""

plt.savefig('cV5 concentration_concentration.tiff',format='tiff',dpi=300)
```

Algorithm to determine the suitable magnitude of velocity between electrolyte tanks (Figure 58**)**

The following algorithm (in python) was developed to calculate the discharge capacity and the capacity loss rate for different values of velocity between electrolyte tanks.

Input files: (COMSOL files) cell potential versus time.

```
import pandas as pd
import numpy as np
from scipy.signal import find_peaks
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
import matplotlib



# creating file names
V_file = ['V0.txt','V6.txt','V7.txt','V8.txt','V9.txt']
v_br_x4 = ['0','1','5','10','50']

CL = []
CL_rate = []

for i in range(len(V_file)):
    df1_raw = pd.read_csv(V_file[i],skiprows=(7))

    df1 = df1_raw.rename(columns={'% X':'time','Height':'V'})



    # selecting data
    x = list(df1['time'])
    y = list(df1['V'])



    #interpolation
    it=200000
    f= interp1d(x,y)
    x_V= np.linspace(np.array(x).min(),np.array(x).max(),it)
    y_V = f(x_V)



    # differentiation of discrete values of V versus time data (for time peak determination)
    dydx_V= np.gradient(y_V,x_V)

    # finding peaks charge/discharge
    peaks = find_peaks(abs(dydx_V),height=0.02)
    height = peaks[1]['peak_heights']
    if height.shape[0] == 1:
```

```
        height =  np.append(peaks[1]['peak_heights'],peaks[1]['peak_heights'])

    peak_pos = np.array(x_V)[peaks[0]]
    if peak_pos.shape[0] == 1:
        peak_pos = np.append(np.array(x_V)[peaks[0]],x_V[len(x_V)-1])


    #calculating capacity loss
    step_time = []
    step_time.append(peak_pos[0])
    z_ite = 0
    for i in range(len(peak_pos)):
        if z_ite == len(peak_pos)-1:
            break
        step_time.append(peak_pos[z_ite+1]-peak_pos[z_ite])
        z_ite += 1
        discharge_time = step_time[1:len(step_time):2]
        capacity_loss = []
        for i in range(len(discharge_time)):
            capacity_loss.append(100*discharge_time[i]/discharge_time[0])

    CL_35 = capacity_loss[0:34]
    CL.append(CL_35)

    CL_pc_list = []
    for i in range(len(CL_35)-1):
        CL_pc = CL_35[i]-CL_35[i+1]
        CL_pc_list.append(CL_pc)
    CL35_rate = np.array(CL_pc_list[0:20]).mean()

    CL_rate.append(CL35_rate)


n_cycles = np.arange(1,35,1)

fig,(ax,ax2) = plt.subplots(1,2)
fig.set_figwidth(10)
plt.subplots_adjust(wspace=0.2)

cmap = matplotlib.cm.get_cmap('viridis')
c=np.linspace(0,20,5)
ax.set_title('(a)')
ax.set_xlabel('Cycle number')
ax.set_ylabel('Discharge capacity (%)')
ax.plot(n_cycles,CL[0],color=cmap(0.0),label='0')
ax.plot(n_cycles,CL[1],color=cmap(0.25),label='1')
ax.plot(n_cycles,CL[2],color=cmap(0.50),label='5')
ax.plot(n_cycles,CL[3],color=cmap(0.75),label='10')
ax.plot(n_cycles,CL[4],color=cmap(1.0),label='50')
ax.legend(title='$v_{p}$ (m/s)')

ax2.set_title('(b)')
ax2.set_xlabel('$v_{p}$ (m/s)')
ax2.set_ylabel('Capacity loss rate (% per cycle)')
ax2.scatter(v_br_x4,CL_rate,c=c,cmap='viridis',marker='D')

plt.savefig('magnitude_test.tiff',format='tiff',dpi=600)
```

Algorithm for regression analysis and 4D plot (Figure 59**)**

The following algorithm (in python) was developed for regression analysis. The variables decodified are used. The purpose was to obtain the coefficients on the same scale for all variables.

Input files: data from **Table 17.** Doehlert design for three variables of Study 2. And **Table 29.** All responses from Doehlert design (Study 2)*.

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm
import matplotlib as mpl


# importing data
base = pd.read_csv('data.txt',sep='\t')
X = base.iloc[:, 0:3].values
y = base.iloc[:, 3].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)

regressor = LinearRegression()
regressor.fit(X1,y)

y_pred = regressor.predict(X1)


### creating data to 2D response surface

mesh = 100
levels = 20
transp = 0.9


v1 = 1
v2 = 5.5
v3 = 10


i1 = 45.36
i2 = 80
i3 = 114.64


c1 = 1.123
c2 = 1.45
c3 = 1.777

v_br = np.linspace(1,10,mesh)
```

```
i_appl = np.linspace(45.36,114.64,mesh)



# v_br versus i_appl com c1
v_br,i_appl=np.meshgrid(v_br,i_appl)

column_=np.linspace(0,0,mesh)
for i1 in range(mesh):

    column = np.array([])
    for i2 in range(mesh):
        prev = np.array([v_br[i1][i2],i_appl[i1][i2],c1]).reshape(1,-1)
        prev_p = poly.fit_transform(prev)
        pred=regressor.predict(prev_p)
        column=np.append(column,pred)
    column_=np.column_stack((column_,column))


Z1 = np.delete(column_,0,1)


# v_br versus i_appl com c2
column_=np.linspace(0,0,mesh)
for i1 in range(mesh):

    column = np.array([])
    for i2 in range(mesh):
        prev = np.array([v_br[i1][i2],i_appl[i1][i2],c2]).reshape(1,-1)
        prev_p = poly.fit_transform(prev)
        pred=regressor.predict(prev_p)
        column=np.append(column,pred)
    column_=np.column_stack((column_,column))


Z2 = np.delete(column_,0,1)


# v_br versus i_appl com c3
column_=np.linspace(0,0,mesh)
for i1 in range(mesh):

    column = np.array([])
    for i2 in range(mesh):
        prev = np.array([v_br[i1][i2],i_appl[i1][i2],c3]).reshape(1,-1)
        prev_p = poly.fit_transform(prev)
        pred=regressor.predict(prev_p)
        column=np.append(column,pred)
    column_=np.column_stack((column_,column))


Z3 = np.delete(column_,0,1)


# parameters


ZQ_min = np.array([round(Z1.min(),2),round(Z2.min(),2),round(Z3.min(),2)]).min()
ZQ_max = np.array([round(Z1.max(),2),round(Z2.max(),2),round(Z3.max(),2)]).max()


### ploting

fig = plt.figure()
ax = fig.gca(projection='3d')

plt.contourf(v_br, i_appl, Z1, levels = levels,alpha=transp,offset=c1)
```

```
plt.clim(0,ZQ_max)

plt.contourf(v_br, i_appl, Z2, levels = levels,alpha=transp,offset=c2)
plt.clim(0,ZQ_max)

plt.contourf(v_br, i_appl, Z3, levels = levels,alpha=transp,offset=c3)
plt.clim(0,ZQ_max)

norm = mpl.colors.Normalize(vmin=ZQ_min,vmax=ZQ_max)
cb4=plt.colorbar(cm.ScalarMappable(norm=norm),
            orientation='vertical',
            ticks=np.arange(0,ZQ_max+0.01,5),
            pad = 0.15)
cb4.set_label('Capacity loss (%)',fontsize=10)
cb4.ax.tick_params(labelsize='small',direction='in')


ax.set(xlabel = '$v_{p}$ (m/s)',ylabel = '$i_{appl}$ (mA/$cm^2$)',zlabel='$c_{act}$ (mol/L)')
ax.xaxis.set_ticks(np.arange(v1, v3+0.01,4))
ax.yaxis.set_ticks(np.arange(45, 115+0.01,30))
ax.zaxis.set_ticks(np.arange(1.15,1.75+0.01,0.3))
ax.set_zlim(c1,c3)
plt.savefig('Fig_name.tiff',format='tiff',dpi=600)
```

Algorithm to calculate optimum velocity between tanks (Figure 60**)**

The following algorithm (in python) was developed to determine the optimum velocity between tanks for any combination of applied current density and concentration of active species.

Input files: data from **Table 17.** Doehlert design for three variables of Study 2. And **Table 29.** All responses from Doehlert design (Study 2)*.

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import numpy as np
import matplotlib


# importing data
base = pd.read_csv('data.txt',sep='\t')
X = base.iloc[:, 0:3].values
y = base.iloc[:, 3].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)

regressor = LinearRegression()
regressor.fit(X1,y)

y_pred = regressor.predict(X1)


### creating data to 2D response surface

mesh = 100
levels = 20
transp = 0.9


i1 = 75
i2 = 45
i3 = 60



c1 = 1.75
c2 = 1.45
c3 = 1.15

v_br = np.linspace(1,10,mesh)
```

```
# case (1)
CL1 = []
for i in range(len(v_br)):
    prev = np.array([v_br[i],i1,c1]).reshape(1,-1)
    prev_p = poly.fit_transform(prev)
    pred=float(regressor.predict(prev_p))
    CL1.append(pred)

# case (2)
CL2 = []
for i in range(len(v_br)):
    prev = np.array([v_br[i],i2,c2]).reshape(1,-1)
    prev_p = poly.fit_transform(prev)
    pred=float(regressor.predict(prev_p))
    CL2.append(pred)

# case (3)
CL3 = []
for i in range(len(v_br)):
    prev = np.array([v_br[i],i3,c3]).reshape(1,-1)
    prev_p = poly.fit_transform(prev)
    pred=float(regressor.predict(prev_p))
    CL3.append(pred)


### finding v optimum

v_opt1=v_br[CL1.index(min(CL1))]
v1_str = str(round(v_opt1,1))
v_opt2=v_br[CL2.index(min(CL2))]
v2_str = str(round(v_opt2,1))
v_opt3=v_br[CL3.index(min(CL3))]
v3_str = str(round(v_opt3,1))


### ploting
cmap = matplotlib.cm.get_cmap('viridis')
fig,ax = plt.subplots()
ax.plot(v_br,CL1,'k-',color=cmap(0),label='case (1) '+'$v_{opt}=$'+v1_str)
ax.plot(v_br,CL2,'r-',color=cmap(0.33),label='case (2) '+'$v_{opt}=$'+v2_str)
ax.plot(v_br,CL3,'b-',color=cmap(1.0),label='case (3) '+'$v_{opt}=$'+v3_str)
ax.set(xlabel = '$v_{p}$ (m/s)',ylabel = 'CL (%)')
plt.legend()

plt.savefig('Optimum_v.tiff',format='tiff',dpi=600)
```

Algorithm for validation of regression analysis (Figure 61**)**

The following algorithm (in python) was developed to validate the regression analysis and plot the **Figure 61**

Input files: validation data.

```
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,r2_score
import matplotlib.pyplot as plt
import numpy as np


# importing data
base = pd.read_csv('data.txt',sep='\t')


X = base.iloc[:, 0:3].values
y = base.iloc[:, 3].values


test1 = pd.read_csv('data_validation_random.txt',sep='\t')
X_test1 = test1.iloc[:,1:4].values
y_test1 = test1.iloc[:,7].values

# setting regression
poly = PolynomialFeatures(degree = 2)
X1 = poly.fit_transform(X)
X1_test1 = poly.fit_transform(X_test1)

regressor = LinearRegression()
regressor.fit(X1,y)

y_pred = regressor.predict(X1)
y_pred_test = regressor.predict(X1_test1)

#Figures of merit
R2_t = r2_score(y_test1,y_pred_test)
R2 = r2_score(y,y_pred)

r2s = str(round(R2,4))
r2ts = str(round(R2_t,4))

fig1,ax1 = plt.subplots()
ax1.plot(np.linspace(0,20,50),np.linspace(0,20,50),'k--')
ax1.plot(y_test1,y_pred_test,'go',markersize=7,label='Test set')
ax1.plot(y,y_pred,'bo',markersize=7,label='Train set')
plt.xlabel('Simulated capacity loss (%)')
plt.ylabel( 'Predicted capacity loss (%)')
plt.text(15,3,'$R^2_{train}$:' + r2s + '\n $R^2_{test}$:' + r2ts)
plt.legend()


plt.savefig('regression.tiff',formate='tiff',dpi=300)
```

Appendix C – Scripts for data treatment (Study 3)

Algorithm to calculate the responses from COMSOL files (Table 32)

The following algorithm (in python) was developed to calculate the voltage efficiency and the overpotential (activation plus concentration) of each designed computational experiment.

Input files: (COMSOL files) cell potential versus SoC for charging and discharging, overpotential (activation plus concentration) versus SoC for charging and discharging.

```
import pandas as pd

file_number=list(range(1,33))

Vd_file = []
Vc_file = []
ec_file = []
ed_file = []

#creating name of files

for i in file_number:
    Vc='Vc' + str(i) +'.txt'
    Vc_file.append(Vc)
    Vd='Vd' + str(i) +'.txt'
    Vd_file.append(Vd)
    ec='ec' + str(i) +'.txt'
    ec_file.append(ec)
    ed='ed' + str(i) +'.txt'
    ed_file.append(ed)

#reading files
Vc_list = []
Vd_list = []
ec_list = []
ed_list = []


for i in range(len(ed_file)):
    df_Vc=pd.read_csv(Vc_file[i],skiprows=(7))
    Vc_mean=df_Vc['Height'].mean()
    Vc_list.append(Vc_mean)
    df_Vd=pd.read_csv(Vd_file[i],skiprows=(7))
    Vd_mean=df_Vd['Height'].mean()
    Vd_list.append(Vd_mean)
    df_ec=pd.read_csv(ec_file[i],skiprows=(7))
    ec_mean=df_ec['Height'].mean()
    ec_list.append(1000*abs(ec_mean))
    df_ed=pd.read_csv(ed_file[i],skiprows=(7))
    ed_mean=df_ed['Height'].mean()
```

```
    ed_list.append(1000*abs(ed_mean))


#calculating voltage efficiency

VE = []
for i in range(len(Vc_list)):
    VE_ite = round(Vd_list[i]*100/Vc_list[i],2)
    VE.append(VE_ite)


#exporting data
treated_data=pd.DataFrame({'VE':VE,'eta_charge':ec_list,'eta_dis':ed_list})
treated_data.to_csv('treated_data.txt',sep='\t',decimal=',',index=False)
```

Algorithm to calculate effects (Figure 65 **and Figure** 67**)**

The following algorithm (in python) was developed to calculate the effects of 2^{5-1} factorial design.

Input files: data from **Table 21** and **Table 32**.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

df=pd.read_csv('effect.txt',sep='\t')

X = df.iloc[:,0:6].values
y = df.iloc[:,6].values

effects = []

# primary effects
list1 = []
list2 = []
list3 = []
list4 = []
list5 = []
list6 = []

for i in range(len(y)):
    it1 = X[i][0]*y[i]
    list1.append(it1)
    it2 = X[i][1]*y[i]
    list2.append(it2)
    it3 = X[i][2]*y[i]
    list3.append(it3)
    it4 = X[i][3]*y[i]
    list4.append(it4)
    it5 = X[i][4]*y[i]
    list5.append(it5)
    it6 = X[i][5]*y[i]
    list6.append(it6)

ef1=2*np.array(list1).mean()
effects.append(ef1)
ef2=2*np.array(list2).mean()
effects.append(ef2)
ef3=2*np.array(list3).mean()
effects.append(ef3)
ef4=2*np.array(list4).mean()
effects.append(ef4)
ef5=2*np.array(list5).mean()
effects.append(ef5)
ef6=2*np.array(list6).mean()
effects.append(ef6)


#secondary effects
list12 = []
```

```
list13 = []
list14 = []
list15 = []
list16 = []
list23 = []
list24 = []
list25 = []
list26 = []
list34 = []
list35 = []
list36 = []
list45 = []
list46 = []
list56 = []


for i in range(len(y)):
    it12 = X[i][0]*X[i][1]*y[i]
    list12.append(it12)
    it13 = X[i][0]*X[i][2]*y[i]
    list13.append(it13)
    it14 = X[i][0]*X[i][3]*y[i]
    list14.append(it14)
    it15 = X[i][0]*X[i][4]*y[i]
    list15.append(it15)
    it16 = X[i][0]*X[i][5]*y[i]
    list16.append(it16)
    it23 = X[i][1]*X[i][2]*y[i]
    list23.append(it23)
    it24 = X[i][1]*X[i][3]*y[i]
    list24.append(it24)
    it25 = X[i][1]*X[i][4]*y[i]
    list25.append(it25)
    it26 = X[i][1]*X[i][5]*y[i]
    list26.append(it26)
    it34 = X[i][2]*X[i][3]*y[i]
    list34.append(it34)
    it35 = X[i][2]*X[i][4]*y[i]
    list35.append(it35)
    it36 = X[i][2]*X[i][5]*y[i]
    list36.append(it36)
    it45 = X[i][3]*X[i][4]*y[i]
    list45.append(it45)
    it46 = X[i][3]*X[i][5]*y[i]
    list46.append(it46)
    it56 = X[i][4]*X[i][5]*y[i]
    list56.append(it56)


ef12=2*np.array(list12).mean()
effects.append(ef12)
ef13=2*np.array(list13).mean()
effects.append(ef13)
ef14=2*np.array(list14).mean()
effects.append(ef14)
ef15=2*np.array(list15).mean()
effects.append(ef15)
ef16=2*np.array(list16).mean()
effects.append(ef16)

ef23=2*np.array(list23).mean()
effects.append(ef23)
ef24=2*np.array(list24).mean()
effects.append(ef24)
ef25=2*np.array(list25).mean()
effects.append(ef25)
ef26=2*np.array(list26).mean()
```

```
effects.append(ef26)

ef34=2*np.array(list34).mean()
effects.append(ef34)
ef35=2*np.array(list35).mean()
effects.append(ef35)
ef36=2*np.array(list36).mean()
effects.append(ef36)

ef45=2*np.array(list45).mean()
effects.append(ef45)
ef46=2*np.array(list46).mean()
effects.append(ef46)

ef56=2*np.array(list56).mean()
effects.append(ef56)



#creating plots

effects_name = ['1','2','3','4','5','6','12','13','14','15','16','23','24',
          '25','26','34','35','36','45','46','56']

df_effects = pd.DataFrame({'effect':effects_name,'Value':effects})

df_effects_or = df_effects.sort_values(by='Value',ignore_index=True)

#porcentage graph
effects_squared = []

for i in range(len(effects)):
   effects_squared.append(abs(effects[i]))

sum_effects_squared = np.array(effects_squared).sum()

percentage_ef = []
for i in range(len(effects_squared)):
   b_ite = 100*effects_squared[i]/sum_effects_squared
   percentage_ef.append(b_ite)

#probability graph and pareto
effect_or = df_effects_or.iloc[:,1]
name_or = df_effects_or.iloc[:,0]

initial = []
initial.append(0)
for i in range(len(effects)-1):
   c_ite = 1/len(effects)+initial[i]
   initial.append(c_ite)

final = []
for i in range(len(initial)):
   final.append(initial[i+1])
   if len(initial) == len(final)+1:
      break
final.append(1)

middle = []
for i in range(len(initial)):
   d_ite = (initial[i]+final[i])/2
   middle.append(d_ite)

inv_norm = norm.ppf(middle)
```

```
#plots
df_perc = pd.DataFrame({'effect':effects_name,'percentage':percentage_ef})
df_perc_or = df_perc.sort_values(by='percentage',ascending=False,ignore_index=True)
percentage_ef = df_perc_or.iloc[:,1].values
name_ef_perc = df_perc_or.iloc[:,0]

fig,ax = plt.subplots()
ax.bar(name_ef_perc,percentage_ef,color='green')
ax.set_xlabel('Standardized Effect')
ax.set_ylabel('Percentage (%)',color='darkgreen')
ax.yaxis.label.set_color('green')
ax.tick_params(axis='y', colors='green')

perc_cum = percentage_ef.cumsum()

ax2 = ax.twinx()
ax2.plot(name_ef_perc,perc_cum,marker='o',color='black')
ax2.set_ylabel('Cumulative Sum (%)',color='black')

#plt.savefig('percentage_graph.png',format='png',dpi=300)

fig3,ax3 = plt.subplots()
ax3.scatter(effect_or,inv_norm,color='green')
ax3.set_xlabel('Effect')
ax3.set_ylabel('z value')
yline = np.linspace(np.array(inv_norm).min(),np.array(inv_norm).max(),10)
xline = np.linspace(0,0,10)
ax3.plot(xline,yline,color='black')

for i, txt in enumerate(name_or):
    ax3.annotate(txt,(effect_or[i],inv_norm[i]))

#plt.savefig('probability_graph.png',format='png',dpi=300)
```