Modelling, Optimization and Control of Energy Systems

Christos Panos

February 2012

Submitted in part fulfilment of the requirements for the degree of Doctor of Philosophy in Chemical Engineering of Imperial College London and the Diploma of Imperial College London
I hereby declare that the work presented in this dissertation is my own and has been otherwise properly acknowledged.

February 2012
Abstract

Multi-parametric programming is a mathematical theory to address optimization problems involving varying parameters. This thesis is concerned with the development of model-based controllers via parametric programming and their application to the design, operation and control of process systems. In part I of this thesis two new algorithms are presented for solving parametric optimization problem of linear state space models via dynamic programming. The first algorithm solves the nominal case, while the second introduce the case of uncertainty in the system matrices. Moreover, an algorithm for robust Explicit model based controller for box-constrained linear systems. These algorithms constitute the basis for the development of model based controllers in the rest of the thesis.

In part II of this thesis, dynamic mathematical models for the cases of metal hydride tank reactor, Proton Exchange Membrane (PEM) fuel cell unit and tunnel kiln process are presented. These mathematical models are used to derive reduced order linear models in order to design explicit/multi-parametric Model Predictive Controllers. Moreover, the extensive design of an experimental PEM fuel cell unit which includes the process and instrumentation diagram (PID), the complete list of materials and the three dimension design of the unit, is presented. Based on the experimental results provided by the manufacturer, a validated dynamic mathematical model of the PEM fuel cell unit is presented. Finally, mathematical modelling, dynamic optimization and design of PI controller for the firing process of a tunnel kiln is presented.
Acknowledgments

I would like to express my gratitude to Professor Stratos Pistikopoulos for his continuous motivation and support during my graduate studies. He has been a continual source of inspiration and supported me with important advices and suggestions and giving me the freedom to develop my ideas. I would also like to thank Professor Nigel Brandon and Nilay Shah for their fruitful comments and help on my work.

The financial support from EPSRC (Projects EP/I014640/1, EP/047017/1), Centre for Process Systems Industrial Consortium and KAUST is gratefully acknowledge.

During my studies I met a number of people that contributed greatly to development of my personality and skills. I am grateful to Nikos Bozinis, Kostas Kouramas for been an inexhaustible source of knowledge and advice through this thesis. I would also like to thank Michael Koutinas, Alexandros Kyparisidis, Matina Zavitsanou and Eleni Pefani. Finally I would like to thank my parents, and my family for the faith and support on my decisions.

This is thesis is dedicated to my mother and my uncle,

Paraskevi and Taso
## Contents

1. Introduction .............................................. 15

I. Theoretical Developments .................................. 24

2. Explicit/multi-parametric MPC of linear discrete-time systems by dynamic and multi-parametric programming 25
   2.1. Introduction .............................................. 25
       2.1.1. Explicit/multi–parametric Model Predictive Control . 28
       2.1.2. Dynamic Programming ............................... 29
   2.2. Algorithm for Explicit/multi–parametric MPC .......... 32
       2.2.1. Dynamic Programming representation of the MPC problem .............................................. 32
       2.2.2. Multi-parametric programming solution strategy for constrained dynamic programming .......... 34
       2.2.3. Novel multi-parametric Model Predictive Control algorithm .............................................. 38
   2.3. Illustrative Examples ..................................... 42
       2.3.1. Example 1 .............................................. 42
       2.3.2. Example 2 .............................................. 45
   2.4. Concluding Remarks ...................................... 46

3. Algorithm for robust explicit/multi-parametric MPC in embedded control systems 48
   3.1. Introduction .............................................. 48
   3.2. Problem Definition ....................................... 51
       3.2.1. Robust multi-parametric programming ............... 53
   3.3. Algorithm for explicit/multi-parametric MPC .......... 56
       3.3.1. Dynamic Programming step - Multi-stage Optimization 56
       3.3.2. Algorithm for explicit/multi–parametric MPC ....... 62
3.4. Examples ......................................................... 65
3.4.1. Example 1 .................................................. 65
3.4.2. Example 2 ................................................... 73
3.5. Concluding Remarks .............................................. 73

4. Robust Explicit/multi-parametric Model Predictive Control for Box-constrained Linear Dynamic Systems 76
4.1. Introduction ...................................................... 76
4.2. A method for Robust Explicit/Multi–parametric MPC ...................................................... 78
  4.2.1. Dynamic programming – multi–stage optimization ...................................................... 78
  4.2.2. Robust Reformulation ....................................... 81
  4.2.3. Multi–parametric procedure ................................ 82
  4.2.4. Algorithm for robust explicit/multi–parametric MPC
        for linear system with “box”–constraints ...................................................... 86
4.3. Concluding Remarks .............................................. 87

II. Energy Systems Application 90

5. Dynamic Optimization and robust explicit model predictive control of hydrogen storage tank 91
5.1. Introduction ...................................................... 91
5.2. Mathematical Model of the Metal-Hydride Bed ...................................................... 93
  5.2.1. Mass balance of hydrogen in the gas phase (bulk of
        the reactor) .................................................... 94
  5.2.2. Pressure drop equations ...................................... 95
  5.2.3. Energy balance equations .................................... 95
  5.2.4. Mass balance of the metal hydride ................................ 96
  5.2.5. Definition of equilibrium pressure ................................ 96
  5.2.6. Modelling the heat exchange process .............................. 97
  5.2.7. Boundary and initials conditions ................................. 97
5.3. Dynamic optimization of hydrogen desorption ...................................................... 98
5.4. Model identification ............................................. 100
5.5. Multi-Parametric Model Predictive Control (mp-MPC) .......................................... 102
5.6. Simulation and validation of the multi-parametric controllers .................................. 104
5.7. Concluding Remarks ............................................. 107
6. Modelling and explicit model predictive control for PEM fuel cell system

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1. PEM Fuel Cell System</td>
<td>108</td>
</tr>
<tr>
<td>6.2. Types of fuel cells</td>
<td>112</td>
</tr>
<tr>
<td>6.3. Basic Operation of a PEM fuel cell</td>
<td>114</td>
</tr>
<tr>
<td>6.4. PEM Fuel Cell Mathematical Model</td>
<td>114</td>
</tr>
<tr>
<td>6.4.1. Cathode Mathematical Model</td>
<td>118</td>
</tr>
<tr>
<td>6.4.2. Anode Mathematical Model</td>
<td>121</td>
</tr>
<tr>
<td>6.4.3. Anode Recirculation</td>
<td>122</td>
</tr>
<tr>
<td>6.4.4. Fuel cell outlet</td>
<td>124</td>
</tr>
<tr>
<td>6.4.5. Membrane Hydration Model</td>
<td>124</td>
</tr>
<tr>
<td>6.4.6. Electrochemistry</td>
<td>126</td>
</tr>
<tr>
<td>6.4.7. Thermodynamic balance</td>
<td>127</td>
</tr>
<tr>
<td>6.4.8. Air Compressor &amp; DC Motor Model</td>
<td>129</td>
</tr>
<tr>
<td>6.4.9. DC motor</td>
<td>130</td>
</tr>
<tr>
<td>6.5. Model Identification</td>
<td>133</td>
</tr>
<tr>
<td>6.6. PEM Fuel Cell Control Design</td>
<td>136</td>
</tr>
<tr>
<td>6.7. Concluding Remarks</td>
<td>137</td>
</tr>
</tbody>
</table>

7. Modelling and explicit model predictive control for PEM fuel cell

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1. Introduction</td>
<td>141</td>
</tr>
<tr>
<td>7.2. Multi-Parametric Model Predictive Control of an Automated Integrated Fuel Cell Testing Unit</td>
<td>142</td>
</tr>
<tr>
<td>7.2.1. Mathematical Modelling and Experimental Setup</td>
<td>142</td>
</tr>
<tr>
<td>7.2.2. Experimental System</td>
<td>143</td>
</tr>
<tr>
<td>7.2.3. Model Identification</td>
<td>144</td>
</tr>
<tr>
<td>7.2.4. Multi-Parametric Model Predictive Controller (mp-MPC)</td>
<td>145</td>
</tr>
<tr>
<td>7.2.5. Simulation Results</td>
<td>146</td>
</tr>
<tr>
<td>7.3. Design of a MPC controller for regulating temperature in PEM fuel cell systems considering stack degradation</td>
<td>149</td>
</tr>
<tr>
<td>7.3.1. System Description</td>
<td>149</td>
</tr>
<tr>
<td>7.3.2. Non-linear PEM fuel cell model</td>
<td>150</td>
</tr>
<tr>
<td>7.3.3. Discrete Linear model</td>
<td>153</td>
</tr>
<tr>
<td>7.3.4. Explicit MPC formulation</td>
<td>156</td>
</tr>
</tbody>
</table>
8. Experimental Unit of PEM Fuel Cell System

8.1. Introduction .................................................. 166
8.2. Equipment ..................................................... 167
  8.2.1. PEM Fuel Cell ........................................... 167
  8.2.2. DC LOAD .................................................. 169
  8.2.3. Transmitters ............................................. 170
  8.2.4. Controllers .............................................. 170
  8.2.5. Valves ................................................... 171
  8.2.6. Pumps .................................................... 172
  8.2.7. Other Equipments ...................................... 173
8.3. Interface BOX (IBX) Description ................................. 173
  8.3.1. Power Relays ............................................. 174
  8.3.2. Indicator Lights ......................................... 174
8.4. Computer Description ........................................ 174
8.5. Safety Management ........................................... 175
8.6. Operation Procedure ........................................ 175
  8.6.1. Heat-up ................................................ 176
  8.6.2. Conditioning .......................................... 176
  8.6.3. Regular operation ...................................... 176
  8.6.4. Shut down ............................................. 176
8.7. Modelling of the PEM Fuel Cell Stack (NEDSTACK 1kW) ... 177
  8.7.1. Mathematical Model .................................... 177
  8.7.2. Nomenclature ......................................... 195
  8.7.3. Parameter Values .................................... 198
  8.7.4. Model Validation ..................................... 199
  8.7.5. Results .............................................. 201
8.8. Concluding Remarks .......................................... 205

9. Modelling, Optimization and Control of a Tunnel Kiln

9.1. Introduction .................................................. 206
9.2. Mathematical Model .......................................... 211
  9.2.1. Mass balances .......................................... 213
  9.2.2. Energy Balances ...................................... 215
9.2.3. Model Validation ........................................ 217
9.2.4. Controller design ........................................ 220
9.2.5. Optimal Kiln Operation ................................. 221
9.2.6. Concluding Remarks ................................... 225

10. Conclusions and Future Research Directions 226
10.1. Conclusions .................................................. 226
10.2. Key contributions of thesis ................................. 228
10.3. Future Work .................................................. 229
  10.3.1. Theoretical and algorithm developments .......... 229
  10.3.2. Multi-parametric Programming ..................... 230
  10.3.3. Explicit nonlinear MPC (mp–NMPC) ................. 230
  10.3.4. Robust multi–parametric/explicit control of hybrid and nonlinear systems ......................... 231
  10.3.5. Design linear feedback gain with additive disturbances 231
  10.3.6. Model reduction techniques ............................. 244
  10.3.7. Applications ............................................. 244
10.4. Publications ................................................ 245
10.5. Acknowledgements ......................................... 247

A. Appendix I-Matlab Code for Explicit Robust MPC by Dynamic Programming 265
  A.1. Code for FinalSolution.m ............................ 265
  A.2. Code for ParametricSolution.m ..................... 269

B. Appendix II-PEM Fuel Cell Lab 281
  B.1. Unit Specifications ..................................... 281
  B.2. List of materials .......................................... 282
  B.3. PEM Fuel Cell unit design .............................. 285
    B.3.1. Frame and Panel .................................. 285
    B.3.2. Control Cabinet .................................. 285
List of Figures

1.1. Overall scheme of the fuel cell system .......................... 16
1.2. Model Predictive Control implementation ........................ 19
1.3. Framework for design explicit/Multi–Parametric MPC ............ 20

2.1. Critical regions for the solution $u_1 = \mu_1(x_1)$ at stage 1 – Example 1 .................................................. 44
2.2. Critical regions for the solution $u_0 = \mu_0(x_0)$ at stage 0 – Example 1 .................................................. 44
2.3. Critical regions for Example 2 ........................................ 46
2.4. State and input time profiles for the system simulation starting at the initial state $x_0 = [-10 \ 10 \ 0 \ 0]^T$ – Example 2 .................................................. 46

3.1. Critical regions of the solution $u_2 = \mu_2(x_2))$ for stage 2 for Example 1 .................................................. 68
3.2. Critical regions of the solution $u_1 = \mu_1(x_1))$ for stage 1 for example 1 .................................................. 69
3.3. Critical regions of the solution $u_0 = \mu_0^*(x_0))$ for stage 0 for Example 1 .................................................. 71
3.4. Projection on the state–space of the critical regions for stage 1 for Example 2 .................................................. 74
3.5. Projection on the state–space of the critical regions for stage 0 for Example 2 .................................................. 75
3.6. Simulation of the states and input for Example 2 ................. 75

5.1. Enhanced metal-hydride bed reactor with heat exchangers (Georgiadis et al., 2009) Georgiadis et al. (2009) ............... 94
5.2. Optimal time profile of $T_{(z=1)}$ ...................................... 99
5.3. Optimal time profile of Hydrogen mass .............................. 100
5.4. Simulation and SS model comparison ................................ 101
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5. Critical Regions of mp-MPC in the $x_1 x_2$ sub-spaces</td>
<td>104</td>
</tr>
<tr>
<td>5.6. Critical regions of mp-MPC in the $x_1 - x_2$ sub-spaces</td>
<td>105</td>
</tr>
<tr>
<td>5.7. Temperature profiles of the mp-MPC, R mp-MPC and PI controllers - nominal conditions</td>
<td>106</td>
</tr>
<tr>
<td>5.8. Hydrogen mass desorbed - nominal conditions</td>
<td>106</td>
</tr>
<tr>
<td>5.9. Temperature profiles of the mp-MPC, robust mp-MPC and PI controllers</td>
<td>107</td>
</tr>
<tr>
<td>6.1. Flows through the T</td>
<td>123</td>
</tr>
<tr>
<td>6.2. Simulation for the stack and coolant temperatures</td>
<td>132</td>
</tr>
<tr>
<td>6.3. Stack voltage</td>
<td>132</td>
</tr>
<tr>
<td>6.4. Different sequences of input variables</td>
<td>135</td>
</tr>
<tr>
<td>6.5. Process and SS model comparison</td>
<td>135</td>
</tr>
<tr>
<td>6.6. Projection on the state-space of the critical regions of mp-MPC</td>
<td>138</td>
</tr>
<tr>
<td>6.7. Stack temperature</td>
<td>138</td>
</tr>
<tr>
<td>6.8. Stack voltage</td>
<td>139</td>
</tr>
<tr>
<td>7.1. Structure of the model</td>
<td>143</td>
</tr>
<tr>
<td>7.2. PEM Fuel Cell Unit</td>
<td>143</td>
</tr>
<tr>
<td>7.3. Control structure</td>
<td>147</td>
</tr>
<tr>
<td>7.4. Temperature control and cooling/heatup</td>
<td>147</td>
</tr>
<tr>
<td>7.5. Control of power and $\lambda O_2$</td>
<td>148</td>
</tr>
<tr>
<td>7.6. Current and mass flow rates</td>
<td>149</td>
</tr>
<tr>
<td>7.7. 1.2 kW Nexa power module</td>
<td>150</td>
</tr>
<tr>
<td>7.8. Heat transfer dynamics validation</td>
<td>153</td>
</tr>
<tr>
<td>7.9. Discrete Linear Model for Temperature prediction</td>
<td>154</td>
</tr>
<tr>
<td>7.10. Discrete Linear Model for Oxygen Excess Ratio prediction</td>
<td>155</td>
</tr>
<tr>
<td>7.11. Critical Regions for Stack temperature set point - Stack temper-</td>
<td>158</td>
</tr>
<tr>
<td>erature</td>
<td></td>
</tr>
<tr>
<td>7.12. Critical Regions for Oxygen excess ratio set point- Oxygen</td>
<td>159</td>
</tr>
<tr>
<td>excess ratio</td>
<td></td>
</tr>
<tr>
<td>7.13. Control architecture</td>
<td>160</td>
</tr>
<tr>
<td>7.14. Control of the fuel cell system scheme</td>
<td>161</td>
</tr>
<tr>
<td>7.15. Stack Temperature Results</td>
<td>162</td>
</tr>
<tr>
<td>7.16. Stack Temperature-Fan Voltage</td>
<td>163</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>7.17. Load Current</td>
<td>164</td>
</tr>
<tr>
<td>7.18. Load Current-Compressor voltage</td>
<td>164</td>
</tr>
<tr>
<td>8.1. Simplified Process Instrumentation Diagram</td>
<td>167</td>
</tr>
<tr>
<td>8.2. Detailed Process Instrumentation Diagram</td>
<td>168</td>
</tr>
<tr>
<td>8.3. Sign convention for streams within the membrane electrolyte</td>
<td>178</td>
</tr>
<tr>
<td>assembly</td>
<td></td>
</tr>
<tr>
<td>8.4. Parameter Estimation of the Fuel Cell Characteristic</td>
<td>200</td>
</tr>
<tr>
<td>8.5. Simulation of a complete operation cycle</td>
<td>202</td>
</tr>
<tr>
<td>8.6. Simulated transport of liquid water within the gas diffusion</td>
<td>204</td>
</tr>
<tr>
<td>layer on the cathode side</td>
<td></td>
</tr>
<tr>
<td>8.7. Simulated molar oxygen concentration within the gas diffusion</td>
<td>204</td>
</tr>
<tr>
<td>layer on the cathode side</td>
<td></td>
</tr>
<tr>
<td>9.1. Kiln operation</td>
<td>208</td>
</tr>
<tr>
<td>9.2. Average gas temperature</td>
<td>209</td>
</tr>
<tr>
<td>9.3. Column layout</td>
<td>213</td>
</tr>
<tr>
<td>9.4. Process and simulation gas temperature</td>
<td>219</td>
</tr>
<tr>
<td>9.5. Simulation Gas and simulation brick temperature</td>
<td>220</td>
</tr>
<tr>
<td>9.6. Simulation PI controller response</td>
<td>221</td>
</tr>
<tr>
<td>9.7. Simulation brick temperature for the PI controller</td>
<td>222</td>
</tr>
<tr>
<td>9.8. Process and optimal gas temperature</td>
<td>223</td>
</tr>
<tr>
<td>9.9. Optimal gas and simulation brick temperature</td>
<td>224</td>
</tr>
<tr>
<td>B.1. 3D Frame Layout</td>
<td>286</td>
</tr>
<tr>
<td>B.2. Frame Layout</td>
<td>286</td>
</tr>
<tr>
<td>B.3. Panel Layout</td>
<td>287</td>
</tr>
<tr>
<td>B.4. Unit Layout</td>
<td>287</td>
</tr>
<tr>
<td>B.5. Unit Layout</td>
<td>288</td>
</tr>
<tr>
<td>B.6. Control cabinet design</td>
<td>288</td>
</tr>
<tr>
<td>B.7. External view of the cabinet</td>
<td>289</td>
</tr>
</tbody>
</table>
List of Tables

2.1. Key publications in Explicit/multi–parametric Model Predictive Control ........................................... 27
2.2. Explicit solution $u_1 = f_1(x_1)$ for stage 1 – Example 1. .... 43
2.3. Explicit solution $u_0 = \mu_0(x_0)$ for stage 0 – Example 1. .... 43
2.4. Execution times and number of parameters for each stage of Algorithm 1 – Example 1. ..................... 45

3.1. Key publications in robust explicit/multi–parametric MPC (mp–MPC) .................................................. 50
3.2. Explicit solution $u_2 = \mu_2^*(x_2)$ for Stage 2 of the Example. .. 67
3.3. Explicit solution $u_1 = \mu_1^*(x_1)$ for Stage 1 of Example 1. .. 70
3.4. Explicit solution $u_1 = f_1^*(x_1, u_2)$ for Stage 1 of the Example. 71
3.5. Explicit solution $u_0 = \mu_0^*(x_0)$ for Stage 0 of Example 1. .. 72
3.6. Execution times and number of parameters for each stage of Algorithm 2 – Example 1. ..................... 72

4.1. Algorithm for Robust Explicit Multi–Parametric MPC ...... 89

6.1. Types of fuel cells .................................................. 113

7.1. Optimization problem parameters and settings ............ 146

8.1. Process Specifications ............................................. 169
8.2. Parameter values of the gas diffusion layer as used in the model 198
8.3. Parameter values of the membrane as used in the model .... 199
8.4. Parameter values of the fuel cell stack as used in the model .. 199
8.5. Parameter values for the stack cooling as used in the model .. 200

9.1. Temperature results for the validation of the firing process .. 218
9.2. Massflowrate results for the validation of the firing process .. 219
9.3. Temperature results for the optimal firing process .......... 224
1. Introduction

Fuel cells are a promising technology for electrical power generation, widely regarded as a potential alternative for stationary and mobile applications. Fuel cells are electrochemical devices that convert the chemical energy of a fuel to electrical energy. The electrical efficiency of the fuel cells is higher than the most conventional devices for power generation, since they avoid intermediate steps of production of mechanical energy. The transport sector is one of the major contributors to global fossil fuel consumption and carbon emissions. Alternative power devices for automotive applications have been actively studied over the last few years with great emphasis on fuel cells. The primary type of fuel cells for automotive industry application is Proton Exchange Membrane (PEM) fuel cells, due to their suitable properties for vehicle applications such as low sensitivity to orientation favourable power to weight ratio and fast and easy start-up.

A typical fuel cell system consists of a PEM fuel cell stack, a compressor, humidifier, a cooling system to maintain the temperature of the stack and a hydrogen storage tank (Figure 1.1). Hydrogen is channelled in the anode side of the fuel cell while air in the cathode side. The compressor and the electric drive motor are used to achieve the desired air mass flow and pressure, while the humidifier has been used to achieve proper humidity of the air in order to minimize the danger of dehydration of the membrane. In addition, a recycling system for hydrogen is usually applied to the system in order to minimize the hydrogen consumption.

Technologies that have been developed for the storage of hydrogen for mobile devices, such as compression and liquefaction (Momirlan and Veziroglu, 2002) are in general impractical due to high energy requirements and space limitations. An alternative technology, using metal-hydride alloys (LaNi5) capable to adsorb and release large amounts of hydrogen by bonding with forming hydrides, has been developed (Jenni et al., 1999) over the years to partially overcome these issues. Modelling, optimization and control of such
systems can play a crucial role in assessing their economic and operability performance, as reliable hydrogen storage alternatives.

![Figure 1.1: Overall scheme of the fuel cell system](image)

If fuel cell systems are to become a credible replacement of existing combustion engines, they must reach a similar level of performance and life time. Without taking into account a manufacturing and material costs, a main technical issue towards this goal are operability issues related to ground vehicle propulsion, which pose a challenging control problem, especially due to the transient behaviour of an integrated fuel cell system such as one shown in Figure 1.1.

Model based control (MPC) strategy is a suitable approach to obtain the optimal operation of the fuel cell system, due to its ability to control multi-input multi-output systems with interactions and disturbances. However, MPC requires an analytical and accurate dynamic mathematical model of the system. The mathematical model should be experimentally validated in several operation conditions, since the controller will be designed on this system.

In model predictive control (MPC), an open-loop optimal control problem is solved at regular intervals (sampling instants), given the current process measurements, to obtain a sequence of the current and future control actions up to a certain time horizon (in a receding horizon control fashion), based on the future predictions of the outputs and/or states obtained by using a mathematical representation of the controlled system. Only the first input of
the control sequence is applied to the system and the procedure is repeated at the next time instant when the new data are available. Being an online constrained optimization method, MPC not only provides the maximum output of the controlled process but also takes into account the various physical and operational constraints of the system.

The benefits of MPC have long been recognized from the viewpoint of cost and efficiency of operations. Nevertheless, its applications maybe restricted due to increased online computational requirements related to the constrained optimization. In order to overcome this drawback, explicit or multi-parametric model predictive control mp-MPC was developed (Pistikopoulos and Morari, 2002b; Pistikopoulos and Dua, 2007a) which avoids the need for repetitive online optimization. In mp-MPC the online optimization problem is solved off-line with multi-parametric programming techniques to obtain the objective function and the control actions as functions of the measured state/outputs (parameters of the process) and the regions in the state/output space where these parameters are valid i.e. as a complete map of the parameters. The control is then applied by means of simple function evaluations instead of typically demanding online optimization computations. The concepts of multi-parametric programming and explicit/multi-parametric MPC control are briefly discussed next.

**Multi-parametric Programming**

Multi-parametric programming is concerned with the solution of the following mathematical problem:

\[
J(x) = \min_u f(u, x) \\
\text{s.t.} \quad h(u, x) = 0 \\
g(u, x) \leq 0 \\
x \in \mathcal{X}
\]  

(1.1)

where \(x\) is the vector of parameters, \(u\) is the vector of optimization variables, \(f\) is a scalar objective function, \(h\) is the vector of equalities typically denoting the model of the system, \(g\) is a vector of constraints, such as lower and upper bounds on \(x\) and \(u\), and \(\mathcal{X}\) is a compact, convex and real set.
The solution of (1.1) has typically the following structure

\[ u(x) = \begin{cases} 
    u^1(x) & \text{if } x \in CR^1 \\
    u^2(x) & \text{if } x \in CR^2 \\
    \vdots & \\
    u^N(x) & \text{if } x \in CR^N 
\end{cases} \]

such that \( CR^1 \cap CR^2 \cap \ldots \cap CR^N \neq \emptyset, i \neq j, \forall i, j = 1, \ldots, N, \) \( CR^i \subseteq X, \ \forall i = 1, \ldots, N \) and \( CR^1 \cup CR^2 \cup \ldots \cup CR^N = X. \) Note that \( CR^i \) denotes a critical region, where a particular optimal solution \( u^i(x) \) is valid. Note that by substituting \( u(x) \) into \( f(u, x) \) provides \( J(x) \), the multi-parametric profile of the objective function.

The procedure for obtaining \( u^i(x) \) and \( CR^i \) depends upon whether the functions \( f, h \) and \( g \) are linear, quadratic, nonlinear, convex, differentiable or not, as well as whether the variables \( u \) and parameter \( x \) are continuous, binary, time-varying or not – i.e. the type of mathematical model involved.

**Multi-parametric/Explicit Model Predictive Control**

Traditional Model Predictive Control (MPC) aims to provide a sequence of control actions/inputs over a future time horizon, that seeks to optimize the controller performance based upon the predicted states of the system. This is achieved by repetitively solving an on-line optimization problem, that describes the (past, present, future) behavior of the system.

MPC problems are typically formulated as the following optimization problem:

\[
\begin{align*}
\min_{u(0), \ldots, u(N_u)} & \quad J = \sum_{k=0}^{N_y} x'(k)Qx(k) + \sum_{k=0}^{N_u} u'(k)Ru(k) \\
\text{s.t.} & \quad x(k+1) = f(x(k), u(k)) \\
& \quad x_{\min} \leq x(k+1) \leq x_{\max}, \ k = 0, 1, \ldots N_c \\
& \quad u_{\min} \leq u(k) \leq u_{\max}, \ k = 0, 1, \ldots N_c
\end{align*}
\]

where \( x \) and \( u \) are the vectors of state and control deviation variables (inputs), respectively; \( N_y, N_u \) and \( N_c \) are the prediction, control and constraint horizons, respectively; \( Q \) and \( R \) are weights on deviations of the state and
control variables, and \( k \) denotes a time interval. The basic idea of the MPC implementation is shown in Figure 1.2, where at the current time interval \( k \), the optimization problem is solved to minimize the state and control deviations from the set point, by implementing the optimal values of the control/input variables. Note that only the first control element is implemented and this sequence is repeated at the next time interval, for the new state measurements or estimates, until the desired or set point values are obtained. The key advantage of MPC is that it is model–based and it can take into account the constraints on the state and control variables. A key limitation is its on–line computational effort, due to the repetitive solution of the underlying optimization problem.

Multivariable programming can be used to obtain \( u \) as an explicit function of \( x \), thereby reducing the online model–based control and optimization problem to a sequence of function evaluations – this is termed multi–parametric or explicit Model Predictive Control (mp–MPC). The parametric profiles in the mp–MPC settings can be stored on a simple computational hardware such as a micro-chip. The concept of replacing the on–line optimization via the exact mapping of its optimal solutions is termed “on–line optimization via off–line optimization”, while the ability of mp–MPC to be implemented on the simplest possible hardware is denoted as “MPC-on-a-chip” technology (Pistikopoulos, 2009).

The key advantages of such an implementation are that (i) it is computationally efficient since it requires simple function evaluations, (ii) it does not require any on-line optimization software, (iii) its explicit form makes mp–MPC ideal for safety critical applications, and (iv) allows for advanced model–based controllers to be implemented in portable and/or embedded
There are cases where the conventional MPC or explicit MPC, cannot guarantee feasibility of the controlled system due to uncertainties. Uncertainties may arise in different case, i) inappropriate selection of the system model, ii) the system has additive disturbance that is unknown, iii) the system state is not perfectly know or the estimation is cannot accurately predict the state values. Then, robust control approach has to be implemented to guarantee constrains satisfaction and optimal control sequence. The first part of the thesis involves theoretical aspects in the area of optimal linear model based control and robust control.

Figure 1.3.: Framework for design explicit/Multi–Parametric MPC

In the second section of the thesis we focus on the mathematical modelling, dynamic optimization and control issues of PEM fuel cell systems according to the framework showing in Figure 1.3. The framework that has been used to obtain and validate the control design consists of four key steps; (Pistikopoulos, 2009),

1. Development of a high fidelity mathematical modelling - used for detailed simulation and (design and operational) optimization studies.

2. Development of a reduced order/approximating model, suitable for multi–parametric MPC

3. Design of nominal & robust multi-parametric MPC controllers
4. Validation of the controllers

Step 1 involves the development of a high-fidelity mathematical model, for performing detailed dynamic simulation and design/operational optimization studies. The model is validated using experimental data in several operation conditions in order to guarantee the accuracy of the simulation results. In step 2, a reduced order approximated model is derived by performing system identification or reduced order techniques on the simulation data. Step 3 corresponds to the design of the multi-parametric/explicit Model Predictive Controllers (mp-MPC), by applying the available theory and tools of multi-parametric programming and control (Pistikopoulos and Dua, 2007b,a). Finally step 4 involves the off-line validation of the derived multi-parametric/explicit controllers.

Thesis Objectives

The main research objectives of this thesis are as follows

1. Modelling optimization and control of the PEM fuel cell system
   a) Modelling and optimization and control of the metal hydride storage tank
   b) Modelling and control of the fuel cell stack

2. Theoretical developments- study on the presence of system uncertainty/variability of the control design
   a) Explicit/multi-parametric MPC of linear discrete-time systems by dynamic and multi-parametric programming
   b) Algorithm for robust explicit/multi-parametric MPC in embedded control systems
   c) Robust Explicit/multi-parametric Model Predictive Control for Box-constrained Linear Dynamic Systems

3. Design an experimental unit of PEM Fuel Cell System

The thesis is organised in the following way. Part I includes the theoretical developments in multiparametric model predictive control. Chapter 2 presents a new algorithm for solving the explicit/multi-parametric Model
Predictive Control (MPC) problem for linear, time invariant discrete time systems, based on dynamic programming and multi–parametric programming techniques. The main algorithm steps are, i) a dynamic programming step, in which the mp–MPC problem is decomposed into a set of smaller sub–problems where only the current control, state variables and constraints are considered, and ii) a multi–parametric programming step in which each sub–problem is solved as a convex multi–parametric programming problem, to derive the control variables as an explicit function of the states. Based on this work on Dynamic Programming (DP), in chapter 3 a new algorithm for robust explicit/multi–parametric Model Predictive Control (MPC) of uncertain, linear discrete-time systems is presented. The proposed algorithm features, i) a DP reformulations of the MPC optimization problem, ii) a robust reformulation of the constraints, and iii) a multi–parametric programming step, where the control variables are obtained as explicit functions of the state variable, such that the state and input constraints are satisfied for all admissible values of the uncertainty. A key feature of the proposed procedure, is that, as opposed to previous methods, it only solves a convex multi–parametric programming problem for each stage of the DP procedure. Finally in chapter 4 an extension to robust explicit MPC algorithm is presented for the case of box constraints in the state and input variables.

Part II of the thesis includes the modelling, optimization and control of the metal hydride reactor and the PEM fuel cell system, as well as the design of a PEM fuel cell unit. In chapter 5 (Part II) we present a detailed mathematical model for the case of desorption process in metal-hydride bed reactors. Design optimization and dynamic simulation and optimization studies are then performed, based on which a reduced order state-space (SS) model, suitable for the design of advanced model-based controllers, is derived. Both nominal and robust mp–MPC controllers are designed, validated and compared to an optimized PI controller. Chapters 6 and 7 present the detailed mathematical modelling of the PEM fuel cell unit and the design of explicit controllers for PEM fuel cell system based on the general framework(1.3). Chapter 8 presents the analytical design of the PEM Fuel Cell Unit for 1kW PEM fuel cell stack from NEDSTACK. This chapter includes the Process Instrumentation Diagram, the list of the materials and the specification of the designed unit. Finally in chapter 9 we describe the mathematical modelling, optimization and PI control of a Tunnel Kiln
unit for the firing process. The dynamic mathematical model was validated through process data of real kiln plant for the entire firing zone. The validated model used to optimise the heat utilization by minimizing the total fuel consumption in the burners while maintain the quality specifications.
Part I.

Theoretical Developments
2. Explicit/multi-parametric MPC of linear discrete-time systems by dynamic and multi-parametric programming

In this chapter we present a new algorithm for solving the explicit/multi-parametric Model Predictive Control (MPC) problem for linear, time-invariant discrete-time systems, based on dynamic programming and multi-parametric programming techniques. The algorithm involves two key steps, i) a dynamic programming step, in which the mp-MPC problem is decomposed into a set of smaller sub-problems where only the current control, state variables and constraints are considered, and ii) a multi-parametric programming step in which each sub-problem is solved as a convex multi-parametric programming problem, to derive the control variables as an explicit function of the states. The key feature of the proposed method, is that it overcomes potential limitations of previous methods for solving multi-parametric programming problems with dynamic programming, such as the need for global optimization for each sub-problem of the dynamic programming step.

2.1. Introduction

Model Predictive Control (MPC) is a control method which aims to derive the sequence of control actions over a finite time horizon, by seeking to optimize the controller performance based on the predicted states of the system under control (Rawlings and Mayne, 2009). This is achieved by repetitively solving an online open-loop control problem at each sampling instant, which minimizes the state and control deviations from the set point based on the prediction of the states given from a mathematical model of the system. Only the first control of the optimal control sequence is
applied to the system and the optimization is repeated in the next time instant for the new values of the states - in that way an implicit feedback controller is obtained. Being a model–based control method and since it relies on online optimization, MPC is a popular method for the control of constrained multivariable systems, where the constraints are explicitly handled in the optimization problem. Nevertheless, the key limitations of MPC is its online computational effort due to the repetitive solution of the underlying optimization and its implicit feedback nature - the optimal values of the control action are determined numerically from the MPC optimization without any knowledge of the governing control laws.

These shortcomings can be overcome by using the explicit/multi–parametric MPC methods (Bemporad et al., 2002; Johansen, 2002; Pistikopoulos and Morari, 2002b; Pistikopoulos and Dua, 2007a). In mp–MPC the online optimization problem involved in traditional MPC is solved with multi–parametric programming methods (Bank et al., 1982; Pistikopoulos and Dua, 2007b) to derive the optimal controls as explicit functions of the system states and the critical regions in the state–space where these functions are valid, hence a feedback control policy is derived which describes the governing control laws of the underlying MPC problem. The online optimization involved in MPC can then be replaced by a sequence of function evaluations, thus reducing the online computational effort. Explicit/multi–parametric MPC has received extensive attention in the control systems literature. An indicative list of key publications is given in Table 2.1 and the interested reader can look in (Pistikopoulos and Dua, 2007b,a; Pistikopoulos, 2009) and references within.

In most available methods for mp–MPC, the main approach for explicit MPC is to solve the underlying MPC optimization problem as a “single–stage” off–line multi–parametric programming problem (Bemporad et al., 2002; Pistikopoulos and Dua, 2007a). It has been shown that, even for the linear cases, the complexity of this multi–parametric program increases with the number of optimization variables, constraints and the prediction horizon (Bemporad et al., 2002; Pistikopoulos and Morari, 2002b; Faísca et al., 2008). In order to overcome this, dynamic programming (DP) techniques (Bellman, 2003; Bertsekas, 2005) have been introduced for solving multi–parametric programming problems such as the explicit/multi–parametric MPC problem. The main idea is to reduce the original multi–stage op-
Table 2.1.: Key publications in Explicit/multi-parametric Model Predictive Control

| Multi–Parametric Model Predictive Control | (Pistikopoulos, 2000; Pistikopoulos and Morari, 2002b; Bemporad et al., 2002) |
| Multi–Parametric Continuous Time Model Predictive Control | (Kojima and Morari, 2004; Sakizlis et al., 2005) |
| Hybrid Multi–Parametric Model Predictive Control | (Bemporad et al., 2000; Sakizlis et al., 2001; Borrelli et al., 2005) |
| Robust Multi–Parametric Model Predictive Control | (Bemporad et al., 2001; Sakizlis et al., 2004; Mayne et al., 2006) |
| Multi–Parametric Dynamic Programming | (Muñoz de la Peña et al., 2004; Borrelli et al., 2005; Fáisca et al., 2008) |
| Multi–Parametric non-Linear Model Predictive Control | (Johansen, 2002; Bemporad, 2003; Dominguez and Pistikopoulos, 2010) |

Optimization problem to a set of smaller “stage” optimization sub-problems where only the control variables, states and constraints at the current stage are considered (Bertsekas, 2005). Dynamic Programming–based methods have been introduced for a number of mp–MPC problems including constrained linear quadratic regulator problem (CLQR) (Muñoz de la Peña et al., 2004; Fáisca et al., 2008), explicit MPC of hybrid systems (Borrelli et al., 2005) and robust explicit MPC (Bemporad et al., 2003).

Although DP can be applied for a wide class of mp–MPC problems, there are certain issues that might affect the complexity of DP. By solving each stage problem in the DP procedure as a multi–parametric programming problem, it generally results into non–linear stage optimization sub–problems (Pistikopoulos and Dua, 2007a). Even for the linear case, the objective functions of each stage optimization becomes piecewise quadratic (Borrelli et al., 2005; Pistikopoulos and Dua, 2007a). Then, either a multi–parametric nonlinear programming problem has to be solved (Pistikopoulos and Dua, 2007a), or a series of multi–parametric Quadratic Programming
(mp–QP) problems at each stage, which however results in the overlapping of critical regions thus requiring the use of comparison techniques, where the objective functions of the problem for each of the (overlapping) critical regions are compared (Borrelli et al., 2005). If the objective function of the MPC optimization problem is linear (i.e. if described by an infinite or one-norm), the comparison is obtained by solving a convex linear program. However, if the objective function of the MPC optimization problem is quadratic, the comparison is in general a non-convex, optimization problem, hence the use of global optimization methods is required (Borrelli et al., 2005; Faisca et al., 2008), which are typically hard to solve.

In this chapter, is presented a new method for explicit/multi-parametric MPC of linear discrete-time systems with state and inputs constraints. The proposed method employs dynamic programming and multi-parametric programming techniques to disassemble the optimization problem of the mp-MPC formulation into a set of smaller stage optimization sub-problems and the optimal control variables for each stage as a function of the systems states. By exploiting basic properties of dynamic programming and the convexity of the optimization problem of the mp-MPC formulation, the proposed algorithm only solves a convex, multi-parametric programming problem at each stage of the DP procedure and hence avoids the need of global optimization.

2.1.1. Explicit/multi-parametric Model Predictive Control

Consider the case of linear, discrete-time time-invariant systems

\[ x_{t+1} = Ax_t + Bu_t \]  

(2.1)

where \( x_t \in \mathcal{X} \subset \mathbb{R}^n \) is the state vector, \( u_t \in \mathcal{U} \in \mathbb{R}^m \) is the system input vector, \( A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m} \) and \( \mathcal{X} \) and \( \mathcal{U} \) are the state and input constraint sets given as follows

\[ \mathcal{X} = \{ x \in \mathbb{R}^n | Gx \leq w \} \]  

(2.2)

\[ \mathcal{U} = \{ u \in \mathbb{R}^m | Mu \leq \mu \} \]  

(2.3)

where \( G \in \mathbb{R}^{n_g \times n}, d \in \mathbb{R}^{n_g}, M \in \mathbb{R}^{m_M \times n}, \mu \in \mathbb{R}^{m_M} \). It is assumed that both sets contain the origin in their interior. The linear, discrete-time
explicit/multi–parametric Model Predictive Control problem is defined as
the following multi–stage optimization problem

\[
V_N^*(x) = \min_{U \in U} J(U, x)
= \min_{U \in U} \sum_{t=0}^{N-1} \{x_t^T Q x_t + u_t^T R u_t\} + x_N^T P x_N
\] (2.4)

s.t. \[x_{t+1} = Ax_t + Bu_t, \quad t = 0, \ldots, N\] (2.5)
\[x_t \in \mathcal{X}, \quad t = 0, \ldots, N\] (2.6)
\[u_t \in \mathcal{U}, \quad t = 0, \ldots, N - 1\] (2.7)
\[x = x_0\] (2.8)

where \(U \in U^{Nm} \subset \mathbb{R}^{Nm}\) is the control sequence of the current and past
control inputs, \(N\) is the prediction horizon and \(Q, P\) and \(R\) are positive
semi-definite and positive-definite matrices respectively of appropriate
dimensions. It is easy to observe from (2.4)–(2.8) that the explicit/multi–
parametric MPC is a multi–stage optimization problem, where the time
instant \(t\) defines each stage of (2.4)–(2.8), and the stage and terminal costs
are quadratic (where \(P\) is usually obtained from solving the Ricatti Equa-
tion for the unconstrained problem (2.4)). The solution of (2.4) obtains an
optimal control sequence \(U^* = \{u_0^*, u_1^*, \ldots, u_{N-1}^*\}\) and the corresponding
optimal path \(x_1^*, x_2^*, \ldots, x_N^*\) which minimizes the stage additive cost in (2.4)
and satisfies the state and input constraints.

Remark 1. In Bemporad et al. (2002); Pistikopoulos and Morari (2002b)
a method was presented for solving (2.4) as a multi–parametric Quadratic
Programming (mp–QP) problem and deriving the control actions \(u_t\) as ex-

clicit functions of the state \(x\), \(u = f(x) = K_i x + c_i\), if \(x \in CR^i\), where \(CR^i\)
is the critical region in the state space in which the control \(u = f(x)\) is valid.

2.1.2. Dynamic Programming

Dynamic Programming (DP) (Bertsekas, 2005) is a well–documented, pow-
erful tool for solving multi–stage problems such as (2.4). Based on the
optimality principle the multistage problem (2.4) can be disassembled in a
set of lower–dimensional problems, where the value function for each stage
of (2.4) is given by (Bertsekas, 2005)

\[
V_t(x_t) = \min_{u_t, \ldots, u_{N-1}} \sum_{k=t}^{N-1} \{x_k^T Q x_k + u_k^T R u_k\} + x_N^T P x_N
\]  

(2.9)

By applying further the optimality condition (Bertsekas, 2005) the multi-stage problem (2.4) can be disassembled in the following recursive optimization problem (Bertsekas, 2005)

\[
V_t(x_t) = \min_{u_t \in U} J_t(u_t, x_t)
\]

\[
= \min_{u_t \in U} \{x_t^T Q x_t + u_t^T R u_t + V_{t+1}(x_{t+1})\}
\]  

(2.10)

s.t. \( x_{t+1} = A x_t + B u_t \)

\( x_t \in \mathcal{X}, \ u_t \in \mathcal{U}, \ x_{t+1} \in \mathcal{X} \)

where \( V_N(x_N) = x_N^T P x_N \). The stage-wise problem (2.10) is solved for each stage \( t = N - 1, \ldots, 0 \) – starting from stage \( N - 1 \) and ending at stage 0 – and only the control inputs \( u_t \) and constraints on the current stage \( t \) are considered. The cost-to-go function \( V_{t+1}(x_{t+1}) \) is the value function for (2.10) for stage \( t+1 \) and is obtained from the state and optimal solutions of the previous stages. Hence, the current optimal solution \( u_t \) of (2.10) depends on the optimal decisions obtained in the previous iterations.

**Remark 2.** Note that \( V^*(x) = V_0(x_0) \) i.e. Dynamic Programming obtains the optimal solution for (2.4) when the last stage \( t = 0 \) is solved (Bellman, 2003; Bertsekas, 2005).

In Muñoz de la Peña et al. (2004); Borrelli et al. (2005); Faísca et al. (2008) a method was proposed for solving the dynamic programming problem (2.10) with multi-parametric programming, to obtain the optimal control decision at each stage \( t \) as an explicit function of its incumbent state \( x_t \)

\[
\begin{align*}
\mu_t(x_t) &= \begin{cases} 
K_1 x_t + c_1^t & \text{if} \ x_t \in \mathcal{CR}^1 \\
\vdots & \vdots \\
K_q x_t + c_q^t & \text{if} \ x_t \in \mathcal{CR}^q 
\end{cases} 
\end{align*}
\]  

(2.11)

where \( \mathcal{CR}^i \) are the critical regions in which the control law \( K_i x_t + c_i^t \), \( i = 1, \ldots, q \) is valid, for all \( t \in [0,1] \) and hence the control sequence
\[ U = \{u_0(x_0), \ldots, u_{N-1}(x_{N-1})\} \]

By using multi-parametric programming to solve each of the stage problems (2.10) the control variables at the current stage are obtained as a piecewise affine function of the states (2.11) and in addition the value function \( V_{t+1}(\cdot) \) as a piecewise quadratic function of the states (Muñoz de la Peña et al., 2004; Borrelli et al., 2005; Faíscia et al., 2008). Hence, the objective function of (2.10) becomes a piecewise quadratic function and thus problem (2.10) becomes a nonlinear multi-parametric optimization problem (Faíscia et al., 2008). For example, the problem (2.10) at stage \( t = N - 1 \) is given as the following optimization problem

\[
V_{N-1}(x_{N-1}) = \min_{u_{N-1} \in U} J_{N-1}(u_{N-1}, x_{N-1}) \\
= \min_{u_{N-1} \in U} \{x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + V_N(x_N)\} \\
\text{s.t. } x_N = A x_{N-1} + B u_{N-1} \\
\hspace{1cm} x_{N-1} \in \mathcal{X}, \ u_{N-1} \in U, \ x_N \in \mathcal{X}
\]

which is a multi-parametric Quadratic Programming (mp–QP) problem with \( u_{N-1} \) as the optimization variable and \( x_{N-1} \) as the problem parameter. The solution to problem (2.12) is a piecewise affine function of the states \( u_{N-1} = f_{N-1}(x_{N-1}) = K_{N-1}^i x_{N-1} + c_{N-1}^i \) and the value function \( V_{N-1}(\cdot) \) is a piecewise quadratic function of the states, i.e. \( V_{N-1}(x_{N-1}) = x_{N-1}^T Q x_{N-1} \). Then, the objective function \( J_{N-2}(u_{N-2}, x_{N-2}) = x_{N-2}^T Q x_{N-2} + u_{N-2}^T R u_{N-2} + V_{N-1}(x_{N-1}) \) of (2.10) at stage \( N - 2 \) is also a piecewise quadratic function. Therefore, the objective function at each stage (2.10) of the dynamic programming procedure is a nonlinear function of the inputs and states of the same stage and the optimization (2.10) at each stage becomes a nonlinear multi-parametric programming problem.

**Remark 3.** Since (2.10) is a nonlinear multi-parametric optimization problem, it is difficult for the available methods for multi-parametric programming to be applied. As it was shown in Muñoz de la Peña et al. (2004); Borrelli et al. (2005), this issue can be addressed by solving at each stage \( t \) as a set of \( q \) simple mp–QP problems, which in general results in overlapping critical regions. Hence, comparison methods have to be employed which result in a, possibly, non-convex optimization problem i.e. a global optimization problem (Faíscia et al., 2008).
The objective here would be to avoid the use of global optimization methods and solve the DP problem and in addition the explicit/multi-parametric MPC problem by only using convex multi-parametric programming methods. Therefore, we present a method for solving the Dynamic Programming problem (2.10) by only solving a convex multi-parametric programming problem for each stage of the Dynamic Programming procedure and thus avoiding the need for global optimization methods.

2.2. Algorithm for Explicit/multi-parametric MPC

The proposed algorithm for solving the explicit/multi-parametric MPC problem (2.4)-(2.8) is realized in two main steps: i) a dynamic programming (DP) step, and ii) a multi-parametric programming step, which are described in the following sections.

2.2.1. Dynamic Programming representation of the MPC problem

The multi-stage problem (2.4)-(2.8) is decomposed into the following stage optimization problems

\[ V_t(x_t) = \min_{u_t \in U} J_t(u_t, x_t) \]

\[ = \min_{u_t \in U} \sum_{i=t}^{N-1} \{ x_i^T Q x_i + u_i^T R u_i \} + x_N^T P x_N \quad (2.13) \]

s.t. \( x_{i+1} = Ax_i + Bu_i \)

\[ x_t \in \mathcal{X}, \quad u_t \in \mathcal{U}, \quad x_{t+1} \in \mathcal{X} \]

for all \( t = N-1, \ldots, 0 \). Problem (2.13) is equivalent to problem (2.9), where only the current control variable is considered in the optimization, since, based on the dynamic programming principle, we assume that the future control variables \( u_{t+1} = \mu_{t+1}(x_{t+1}), \ldots, u_{N-1} = \mu_{N-1}(x_{N-1}) \) have obtained in the previous stages of dynamic programming. In addition, problem (2.9) is equivalent to (2.10), where the previous solutions \( u_{t+1} = \mu_{t+1}(x_{t+1}), \ldots, u_{N-1} = \mu_{N-1}(x_{N-1}) \) have not yet been substituted in the cost-to-go.
function function $V_{t+1}(x_{t+1})$.

**Remark 4.** The stage-wise optimization problems (2.13) is convex with respect to the control variables $u_t, \ldots, u_{N-1}$ and the state variable $x_t$ since $Q \geq 0$ and $R > 0$.

The convexity of problem (2.13) can then be used to derive an algorithm that solves each of the stage problem (2.13) (and hence (2.10)) as convex multi–parametric programming problems. More specifically, by

1. substituting the prediction of the state $x_i = A^i x + \sum_{j=0}^{i-1} A^{i-j} B u_j$ for all $i = t+1, \ldots, N-1$ in the objective function and inequalities of (2.13), and

2. considering $u_t$ as the optimization variable, and $\theta = [x_t \ u_t \ldots u_{N-1}]^T$ the vector of parameters for (2.13)

a multi–parametric Quadratic Programming (mp-QP) reformulation of the problem (2.13) is obtained as follows ((Pistikopoulos and Morari, 2002b; Faisca et al., 2008))

$$V_t(x_t) = \min_{u_t \in U} \left\{ \frac{1}{2} u_t^T H u_t + \theta^T F u_t \right\} + \theta^T Y \theta$$

(2.14)

s.t.

$$G A x_t + G B u_t \leq w$$

$$G x_t \leq w \quad M u_t \leq \mu_t$$

where the matrices $H, F, Y$ are obtained after substituting the system’s linear model into the objective function and are of appropriate dimensions and $H > 0$.

**Remark 5.** Note that if traditional DP methods had been applied, the values of control variables $u_{t+1}, \ldots, u_{N-1}$, obtained at the previous stages of the DP procedure, would have been substituted in (2.14), hence leading to a non–linear multi–parametric programming problem with a piecewise quadratic function, as it was described in section 2 and remark 5, that would have needed the use of global optimization methods.

Nevertheless, (2.14) is a convex mp-QP problem, since $H > 0$ (2.14) which can be solved by applying conventional multi–parametric programming methods (Pistikopoulos, 2009). The solution of problem (2.14) as an mp–QP problem is presented in the following section.
2.2.2. Multi-parametric programming solution strategy for constrained dynamic programming

Since (2.14) is a convex mp-QP problem, its solution is given by the following explicit linear piecewise affine (PWA) expression (Pistikopoulos and Dua (2007b,a))

$$u_t = f_t(\theta_t) = \begin{cases} 
K_i^t \theta_t + c_i^t & \text{if } \theta_t \in \mathcal{CR}_i^t \\
\vdots & \\
K_s^t \theta_t + c_s^t & \text{if } \theta_t \in \mathcal{CR}_s^t 
\end{cases} \quad (2.15)$$

where $\mathcal{CR}_i^t = \{\theta_t \mid A_i^t \theta_t \leq b_i^t\}$

where $K_i^t, c_i^t$, are of appropriate dimensions and $\mathcal{CR}_i^t, i = 1, \ldots, s_t$ are the critical regions where the linear affine control laws $K_i^t \theta_t + c_i^t$ are valid. Note from (2.15) that the control variable $u_t$ at the current stage $t$ is an explicit function of the state $x_t$ at the current stage of the DP problem but also of the future control variables.

**Remark 6.** Recall, that by using DP methods to solve the explicit MPC problem, the value of the control $u_t$ derived at each stage $t$ of the DP procedure depends on the values of the control inputs $u_{t+1}, \ldots, u_{N-1}$ derived in the previous stages by means of the cost–to–go function (Bertsekas (2005)). This is also expressed in the relation (2.15), where instead of replacing the controls $u_{t+1}, \ldots, u_{N-1}$ with their values, we considered them as parameters for the stage problem (2.14).

Expression (2.15) described the control variable $u_t$ as an explicit function of $x_t, u_{t+1}, \ldots, u_{N-1}$. However, since the control inputs $u_{t+1}, \ldots, u_{N-1}$ have been already obtained from the previous stages of the DP procedure, we can use them to eliminate $u_{t+1}, \ldots, u_{N-1}$ from the expression (2.15) and derive $u_t$ only as an explicit function of the the state $x_t$ i.e. derive a state-feedback solution. We present next a procedure for obtaining $u_t = \mu_t(x_t)$, which for simplicity will be first demonstrated for stages $N-1$ and $N-2$ and it will be then extended for every stage $t$.

**Stage $N-1$:** In stage $N-1$ the control variable is $u_{N-1}$ and the parameter is only the state $x_{N-1}$. Therefore the solution of (2.14) for stage $N-1$ is
obtained from (2.15) by using $t = N - 1$ and $\theta_{N-1} = x_{N-1}$

$$u_{N-1} = f_{N-1}(x_{N-1}) = K^i_{N-1}x_{N-1} + c^i_{N-1} \quad \text{if } x_{N-1} \in CR^i_{N-1}, \ i = 1, \ldots, s_{N-1}$$

where $CR^i_{N-1} = \{x_{N-1} | A^i_{N-1}x_{N-1} \leq b^i_{N-1}\}$

i.e. $u_{N-1}$ is already an explicit PWA function of its incumbent state $x_{N-1}$.

**Stage N−2:** In stage $N−2$ the optimization variable for (2.14) is $u_{N−2}$ and the parameter vector is $\theta_{N-2} = [x_{N-2} \ u_{N-1}]^T$. The solution of the mp-QP problem (2.14) for stage $N−2$ is then given by

$$u_{N-2} = f_{N-2}(x_{N-2}) = K^j_{N-2}x_{N-2} + L^j_{N-2}u_{N-1} + c^j_{N-2} \quad \text{if } x_{N-2}, u_{N-1} \in CR^j_{N-2}, \ j = 1, \ldots, s_{N-2}$$

where

$$CR^j_{N-2} = \{x_{N-2} | A^j_{N-2}x_{N-2} + B^j_{N-2}u_{N-1} \leq b^j_{N-2}\}$$

i.e. $u_{N-2}$ is an explicit expression of both $x_{N-2}$ and $u_{N-1}$. Substituting $x_{N-1} = Ax_{N-2} + Bu_{N-2}$ in (2.16), we derive $u_{N-1} = f_{N-1}(x_{N-2}, u_{N-2})$. Then, the expressions (2.16) and (2.17) are combined to obtain the following set of PWA expressions on $x_{N-2}, u_{N-2}$ and $u_{N-1}$

$$u_{N-1} = K^i_{N-1}Ax_{N-2} + K^i_{N-1}Bu_{N-2} + c^i_{N-1}$$

$$u_{N-2} = K^j_{N-2}x_{N-2} + L^j_{N-2}u_{N-1} + c^j_{N-2} \quad \text{if } x_{N-2}, u_{N-2} \in CR^j_{N-2}, \ x_{N-2}, u_{N-1} \in CR^i_{N-1}$$

(2.18) (2.19)

(2.20)

The control $u_{N-2}$ can then be derived as an explicit function of $x_{N-2}$ by eliminating $u_{N-1}$ either by i) by substituting $u_{N-1}$ from (2.18) into (2.19)-(2.20) and then solving for $u_{N-2}$, or ii) by using elimination methods such as orthogonal projection or Fourier-Motzkin elimination. Both operations are linear, and hence when performed on the set of linear equalities and inequalities (2.18)-(2.20), the resulting expression for $u_{N-2}$ is a also linear
PWA expression of \(x_{N-2}\) (Faísca et al. (2008))

\[
u_{N-2} = \mu_{N-2}(x_{N-2}) = K_{N-2}^i x_{N-2} + c_{N-2}^i \quad (2.21)
\]

if \(x_{N-2} \in CR_{N-2}^i, \ i = 1, \ldots, s_{N-2}\)

where \(CR_{N-2}^i = \{x_{N-2} \mid A_{N-2}^i x_{N-2} \leq b_{N-2}^i \}\)

**Remark 7.** Note that (2.18)-(2.20) are obtained for all possible combinations \(i, j\) of control expressions and critical regions from (2.16) and (2.17). Each of these combinations corresponds to a set of values of \(x_{N-2}, u_{N-2}\) and \(u_{N-1}\), described by (2.18)-(2.20).

**Remark 8.** If a combination (2.18)-(2.20) is feasible i.e. the resulting set of \(x_{N-2}, u_{N-2}\) and \(u_{N-1}\) in (2.18)-(2.20) is not empty, then the corresponding control sequence \(\{u_{N-2}, u_{N-1}\}\) is a feasible solution to (2.14) and hence to (2.13).

**Remark 9.** It is possible that not all combinations of (2.16) and (2.17) are realisable and the set of \(x_{N-2}, u_{N-2}\) and \(u_{N-1}\) in (2.18)-(2.20) is empty, which further implies that no feasible control exists for (2.13). The combinations corresponding to an empty set in (2.18)-(2.20) are then removed and only the ones corresponding to non-empty sets are kept.

The same procedure can be applied for any stage \(t\). At any stage \(t\) the optimal control for (2.13) is given by (2.15) and the solutions of the previous stages are given by

\[
u_{t+1} = f_{t+1}(x_{t+1}) \ldots u_{N-1} = f_{N-1}(x_{N-1}) \quad (2.22)
\]

The linear system model \(x_i = A^{i-t} x_t + \sum_{j=t}^{i-1} A^{i-1-j} B u_j\) is then substituted in (2.22) for all \(i = t + 1, \ldots, N - 1\) to express all equalities in (2.22) in terms of \(x_t, u_t, \ldots, u_{N-2}\). Then, (2.15) and (2.22) are combined to obtain
the following set of PWA expressions

\[ u_{N-1} = f_{N-1}(x_t, u_t, \ldots, u_{N-2}) \] (2.23)

\[ \vdots \]

\[ u_{t+2} = f_{t+2}(x_t, u_t, u_{t+1}) \] (2.24)

\[ u_{t+1} = f_{t+1}(x_t, u_t) \] (2.25)

\[ u_t = f_t(x_t, u_{t+1}, \ldots, u_{N-1}) \] (2.26)

An explicit control \( u_t = \mu_t(x_t) \) as an explicit function of \( x_t \) can be derived from (2.23)-(2.26), if the control variables of the previous stages \( u_{t+1}, \ldots, u_{N-1} \) are eliminated by

1. either substituting (2.23)-(2.25) into (2.26) and then solving with respect to \( u_t \) and \( x_t \), or

2. by using elimination methods such as orthogonal projection or Fourier-Motzkin elimination on the combined linear piecewise affine expressions (2.23)-(2.26).

As it was shown for the case of stage \( N-2 \), the resulting control \( u_t \) will be give as a function of the state \( x_t \)

\[ u_t = \mu_t(x_t) = K^t_i x_t + c^t_i, \text{ if } x_t \in CR^t_i \] (2.27)

where \( CR^t_i = \{ x_t | A^i x_t \leq b^i \} \) (2.28)

**Remark 10.** Note that in the proposed procedure the control variables of the previous stages (2.23)-(2.25) are not substituted directly in (2.13) and then (2.13) is solved. Instead (2.13) is solved first as a convex multi-parametric program and then (2.23)-(2.25) are substituted in the solution of (2.14) to derive the final solution (2.27). This allows for (2.14) to be solved as a convex mp-QP problem and avoid global or combinatorial optimization.

**Remark 11.** The set of feasible parameters (i.e. the states for which a feasible control for (2.14) exist) for (2.13) at each stage \( t \) can be obtained from (2.27) as \( X_t = \cup_{i=0}^{n_t} CR^t_i \). In addition, the set of feasible parameters for (2.13) for \( t = 0 \) is the set \( X_0 \) which also implies that \( X_0 \) is the set of feasible states for (2.4).
2.2.3. Novel multi-parametric Model Predictive Control algorithm

Based on the procedure described in sections 2.2.1 and 2.2.2, we can now present an algorithm for the solution of the explicit MPC problem. The steps of the proposed algorithm are given in detail in Algorithm 1. The algorithm starts at stage \( t = N - 1 \), and iterates through Steps 2-6 of Algorithm 2 for each stages \( t \), until \( t = 0 \). In Step 3 the mp-QP problem (2.14) is solved for the current stage \( t \) to derive \( u_t = f_t(\theta_t) \) as a function of the state and future control variables. Then, in Step 4 the elimination procedure described in Section 2.2.2 is applied to obtain the explicit control \( u_t = \mu_t(x_t) \). After the algorithm terminates at the last stage \( t = 0 \), a sequence of explicit feedback control laws \( U = \{\mu_0(x_0), \ldots, \mu_{N-1}(x_{N-1})\} \) is computed, where each control law \( \mu_t(x_t) \) is given by a PWA expression (2.27).

We now recall the following result from Faísca et al. (2008)

**Lemma 1** (Faísca et al. (2008)). If \( x_t, u_t^*(x_t), \ldots, u_{N-1}^*(x_{N-1}) \) is the solution for each stage \( t \) of the Dynamic Programming problem (2.13) then the following inequalities are satisfied

\[
V_t(x_t, u_t^*(x_t), u_{t+1}^*(x_{t+1}), \ldots, u_{N-1}^*(x_{N-1})) \\
\leq V_t(x_t, u_t(x_t), u_{t+1}^*(x_{t+1}), \ldots, u_{N-1}^*(x_{N-1}))
\]

\[
V_{t+1}(x_{t+1}, u_{t+1}(x_{t+1}), \ldots, u_{N-1}^*(x_{N-1})) \\
\leq V_{t+1}(x_{t+1}, u_{t+1}(x_{t+1}), \ldots, u_{N-1}^*(x_{N-1}))
\]

**Proof** The proof of the Lemma is given in Faísca et al. (2008) and results from the convexity of problem and the properties of the DP procedure.

The above Lemma shows that the control sequence \( u_t^*(x_t), \ldots, u_{N-1}^*(x_{N-1}) \) obtained at each stage \( t \) of the dynamic programming procedure is optimal for all stages \( i \geq t + 1 \). The following result is a direct consequence of Lemma 1

**Theorem 2.2.1.** The optimal control sequence \( U = \{\mu_0(x_0), \ldots, \mu_{N-1}(x_{N-1})\} \) obtained by Algorithm 2, is an optimal control sequence for the explicit/multi-parametric MPC problem (2.4).
Proof Each control law $\mu_t(x_t)$ satisfies the state and input constraints of (2.13) and hence the state and input constraints of (2.4), therefore $U = \{\mu_0(x_0), \ldots, \mu_{N-1}(x_{N-1})\}$ is a feasible solution for (2.4). Moreover, from Lemma 1, we have that $U$ is an optimal solution for (2.13) for all $t \geq 0$. Since $V_0(x_0) = V^*_N(x_0)$, then $U$ is also an optimal solution for (2.4).

The above theorem shows that Algorithm 1 obtains the explicit solution to the explicit MPC problem (2.4).

Algorithm 1 Algorithm for explicit/multi–parametric MPC

Data: $A, B, Q, R, P, N, X, U, T$

Output: A control sequence of explicit control laws $\{u_t, \ldots, u_{N-1}\}$ for the explicit/multi-parametric MPC problem

1: Set $t = N - 1$ (start at stage $N - 1$)
2: Repeat for each stage $t$
3: Solve (2.14) with $u_t$ being the optimization variable and $\theta_t = \{x_t, u_{t+1}, \ldots, u_{N-1}\}$ being the parameter vector. Obtain $u_t = f_t(\theta_t)$ from (2.15).
4: Eliminate $u_{t+1}, \ldots, u_{N-1}$ to obtain $u_t = \mu_t(x_t)$ (Equation (2.27))
5: Set $t = t - 1$
6: Until final stage $t = 0$ has been reached

The main characteristic and advantages of the proposed procedure are now listed next:

- the proposed procedure (Algorithm 2) solves the explicit/multi-parametric MPC problem (2.4), based on the solution of the Dynamic Program-
ming via multi-parametric programming,

- the solution of the multi-stage optimization (2.4) is replaced by solving a sequence of problems of smaller dimensions (2.14) i.e. with a smaller number of optimization variables and a smaller number of constraints,

- Algorithm 1 derives the sequence of optimal control laws $U = \{\mu_0(x_0), \ldots, \mu_{N-1}(x_{N-1})\}$ for the explicit/multi-parametric MPC problem (2.4) (see Theorem 2.2.1),

- at each stage of the proposed Algorithm only a convex multi-parametric Quadratic Programming problem (2.14) (Step 3 of Algorithm 1) is solved and only linear algebraic manipulations are used (Step 4 of Algorithm 1) to obtain the optimal solution.

Implementation: The optimal control sequence $U$ is implemented to the system in an MPC fashion i.e. by only applying the first control $u_0 = \mu_0(x_0)$ of the sequence. In that way an explicit/multi-parametric controller is obtained

$$u(x) = \mu_0(x)$$

which consists of a linear PWA state feedback control law. Alternatively, the whole optimal control sequence $U$ can be applied in an open-loop control fashion, by applying sequentially each of the controls $u_t = \mu_t(x_t)$ at each time instant $t$, although this will result in the loss of the prediction and feedback properties of the MPC controller.

Computational Complexity: An analysis of the computational complexity of the proposed procedure and a comparison with the methods presented in Pistikopoulos and Morari (2002b); Bemporad et al. (2002) which solve (2.4) as a single mp-QP problem, is provided next. It was shown in Pistikopoulos and Morari (2002b); Pistikopoulos and Dua (2007a) that if (2.4) is solved as a single multi-stage mp-QP problem, then an upper bound to the number $\eta_{CR}$ of critical regions of the explicit solution to (2.4) is given by

$$\eta_{CR}^{mpQP} \leq \sum_{k=0}^{n-1} k!\cdot (m_M + n_g)^k$$
where
\[
\eta = \sum_{i=0}^{N \cdot m} \frac{N \cdot (m_M + n_g)!}{(N \cdot m_M + N \cdot n_g - i)! i!}
\]

Note that this upper bound on the number of critical regions depends on the prediction horizon \(N\). This implies that the number of critical regions may increase significantly if a large prediction horizon is used.

Following similar arguments in this proposed approach, we can derive an upper bound for each of the stage problems (2.14). Since in (2.14), the number of optimization variables for each stage optimization (2.13) is \(m\), the number of parameters is \(n + (N - t - 1) \cdot m\) and the number of linear inequality constraints is \(m_M + n_g\), an upper bound to the number of critical regions for (2.14) is given by

\[
\eta_{ineq}^{DP} \leq \sum_{k=0}^{\eta-1} k! (m_M + n_g)^k \text{ where } \eta = \sum_{i=0}^{m} \frac{(m_M + n_g)!}{(m_M + n_g - i)! i!}
\]

The above relations shows that the number of critical regions for stage-wise optimizations (2.13) does not depend on the prediction horizon \(N\) and hence the complexity of the stage-wise problems remain the same even if \(N\) increases. This implies that even if the multi-stage problem (2.4) becomes harder or even intractable to solve for large \(N\), solving each one of the smaller problems (2.13) will still remain a much more tractable procedure.

On the other hand, the number of parameters at each stage increases since the number of future inputs considered in each stage increases. Hence, the number of parameters that have to be eliminated in (2.15) and (2.23)-(2.26) increases, in order to derive the explicit solution (2.27) and thus affecting the computational complexity of the algebraic computations in (2.23)-(2.26) that are needed to derive the explicit control (2.27). Nevertheless, the operations performed at this step of the proposed algorithm rely only on linear operations such as Fourier-Motzkin or Gauss elimination (see Schrijver (1986)).
2.3. Illustrative Examples

2.3.1. Example 1

The proposed methodology is first illustrated for the following example, (taken from Bemporad et al. (2002))

\[ A = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix}, \quad N = 2 \]

\[ Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad P = \begin{bmatrix} 1.8588 & 1.2899 \\ 1.2899 & 6.7864 \end{bmatrix}, \quad R = 0.01 \]

\[ \mathcal{X} = \mathbb{R}^2, \quad \mathcal{U} = \{ u \in \mathbb{R} \mid -2 \leq u \leq 2 \} \quad (2.29) \]

Algorithm 1 is applied for the above problem and each of the steps are described below in detail. The algorithm starts at stage \( N - 1 = 1 \)

**Stage 1:** In step 3 of Algorithm 1, the stage optimization problem \( V_1(x_1) = \min_{u_1 \in \mathcal{U}} \{ x_1^T Q x_1 + u_1^T R u_1 + x_2^T P x_2 \} \)

\[ \text{s.t. } x_2 = A x_1 + B u_1, \quad -2 \leq u_1 \leq 2 \]

is solved as an mp-QP problem with \( u_1 \) being the optimization variable and \( x_1 \) being the parameter. The mp–QP algorithm of Pistikopoulos and Morari (2002b) is used to derive the explicit solution \( u_1 = \mu_1(x_1) \) which is given in Table 2 and Figure 2.1. Since \( u_1 \) is already a function of \( x_1 \) there is no need to implement step 4 of the algorithm.

**Stage 0:** Step 3 of Algorithm 1 is applied and the problem

\[ V_0(x_0) = \min_{u_0 \in \mathcal{U}} \{ x_0^T Q x_0 + u_0^T R u_0 + x_1^T Q x_1 + u_1^T R u_1 + x_2^T P x_2 \} \]

\[ \text{s.t. } x_2 = A x_1 + B u_1, \quad x_1 = A x_0 + B u_0 \]

\[ -2 \leq u_0 \leq 2 \quad (2.30) \]

is solved as an mp–QP with \( u_0 \) being the optimization variable and \( x_0, u_1 \) are the parameters. The explicit solution of (2.30) is a PWA function \( u_0 = f_0(x_0, u_1) \) of \( x_0, u_1 \) that consists of 3 critical regions - due to space
Table 2.2.: Explicit solution \( u_1 = f_1(x_1) \) for stage 1 – Example 1.

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ( u_1 = 2 )</td>
<td>( 0.9961x_1^1 + x_1^2 \leq -0.3292 )</td>
</tr>
<tr>
<td>2 ( u_1 = -6.051x_1^1 - 6.074x_1^2 )</td>
<td>(-0.9961x_1^1 - x_1^2 \leq 0.3292) (+0.9961x_1^1 + x_1^2 \leq 0.3292)</td>
</tr>
<tr>
<td>3 ( u_1 = -2 )</td>
<td>(-0.9961x_1^1 - x_1^2 \leq -0.3292 )</td>
</tr>
</tbody>
</table>

Table 2.3.: Explicit solution \( u_0 = \mu_0(x_0) \) for stage 0 – Example 1.

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ( u_0 = 2 )</td>
<td>( 0.9961x_0^1 + x_0^2 \leq -0.5119 )</td>
</tr>
<tr>
<td>2 ( u_0 = 2 )</td>
<td>(-0.9961x_0^1 - x_0^2 \leq 0.5119) (+0.9961x_0^1 + x_0^2 \leq -0.3292)</td>
</tr>
<tr>
<td>3 ( u_0 = -6.051x_0^1 - 6.074x_0^2 )</td>
<td>(-0.9961x_0^1 - x_0^2 \leq 0.3292) (+0.9961x_0^1 + x_0^2 \leq 0.3292)</td>
</tr>
<tr>
<td>4 ( u_0 = -2 )</td>
<td>(-0.9961x_0^1 - x_0^2 \leq -0.3292) (+0.9961x_0^1 + x_0^2 \leq 0.5119)</td>
</tr>
<tr>
<td>5 ( u_0 = -2 )</td>
<td>(-0.9961x_0^1 - x_0^2 \leq -0.5119 )</td>
</tr>
</tbody>
</table>

limitations we present here the mathematical representation of the second critical regions and corresponding control law

\[
 u_0 = -7.104x_0^1 - 7.131x_0^2 - 0.3494u_1, \\
 \text{if } x_0, u_1 \in CR^2_0 = \{ -0.9962x_0^1 - x_0^2 - 0.049u_1 \leq 0.2805, +0.9962x_0^1 + x_0^2 + 0.049u_1 \leq 0.2805 \} \quad (2.31)
\]

Then, in step 4 of Algorithm 1, the system model \( x_1 = Ax_0 + Bu_0 \) is incorporated in the expression of Table 2 to obtain \( u_1 = f_1(x_0, u_0) \). The new expression \( u_1 = f_1(x_0, u_0) \) is incorporated in the piecewise affine expressions of (2.31) to form equations (2.18)–(2.19) and \( u_1 \) is eliminated from the expressions of \( u_0 \) by direct substitution to obtain the final solution \( u_0 = \mu_0(x_0) \). The final solution is given in Table 2.3 and Figure 2.2. Since only the first control will be applied to the system, the solution to the explicit/multi-parametric MPC problem in this example is the explicit control law \( u_0 = \mu_0(x_0) \). Note that the solution \( u_0 = \mu_0(x_0) \) obtained with the proposed algorithm, is the same as the solution obtained with the method of (Bemporad et al., 2002) where the explicit/multi-parametric MPC problem is solved as a single mp–QP problem.
Finally, Table 2.4 presents the execution times and number of parameters for an indicative number of stages of algorithm 1, when the horizon for problem (3.31) is $N = 50$. The total execution time for algorithm 1 is 433.2 sec. Note that the number of parameters for each stage increases and hence the execution time for each stage increases (due to the increasing number of parameters that have to be eliminated in each stage). Nevertheless, the execution time for each stage is not significant, and the algorithm can be extended to larger horizons. It is also expected that the execution time will increase for large values of the system inputs and and constraints.
Table 2.4: Execution times and number of parameters for each stage of Algorithm 1 – Example 1.

<table>
<thead>
<tr>
<th>Stages</th>
<th>49</th>
<th>48</th>
<th>26</th>
<th>25</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0.01</td>
<td>0.109</td>
<td>1.61</td>
<td>1.70</td>
<td>7.47</td>
<td>6.26</td>
</tr>
<tr>
<td># of parameters</td>
<td>1</td>
<td>2</td>
<td>24</td>
<td>25</td>
<td>49</td>
<td>50</td>
</tr>
</tbody>
</table>

2.3.2. Example 2

We now consider the linear, discrete–time system (3.8) with

\[
A = \begin{bmatrix}
0.9233 & 0 & 0.1812 & 0 \\
0 & 0.9461 & 0 & 0.1492 \\
0 & 0 & 0.8111 & 0 \\
0 & 0 & 0 & 0.8464 \\
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
0.40 & 0.0227 \\
0.012 & 0.3055 \\
0 & 0.2159 \\
0.1437 & 0 \\
\end{bmatrix}
\]

(2.32)

with state and input constraints

\[
\mathcal{X} = \{x \in \mathbb{R}^4 | -10 \leq x_i \leq 10, \ i = 1, \ldots, 4\}
\]

\[
\mathcal{U} = \{u \in \mathbb{R}^2 | -1 \leq u_i \leq 1, \ i = 1, 2\}
\]

The explicit/multi–parametric MPC (2.4) is then formulated with \(Q = I_4\), \(R = 0.01 \cdot I_2\), \(P = 10 \cdot I_4\) and prediction horizon \(N = 3\), where \(I_n\) denotes the \(n \times n\) identity matrix. By applying Algorithm 1 for this problem we obtain an explicit MPC controller which consists of 661 critical regions and corresponding affine control laws - note that the critical regions of the explicit MPC are given at stage 0 of Algorithm 1. A two dimensional projection of the critical regions on the space of states \(x_1\) and \(x_2\) is given in Figure 2.3. Finally, Figure 2.4 shows the time profiles of the four system states and two system inputs.
2.4. Concluding Remarks

In this chapter, we presented a method for solving the explicit/multi-parametric MPC problem based on the solution of Dynamic Programming with multi-parametric programming. The method features a two step algorithm which, based on DP methods, decomposes the explicit/multi-parametric MPC problem into a set of smaller stage optimization problems and solves each of these stage problem as simple convex multi-parametric programming.
problem. The key advantage of the proposed algorithm is that it overcomes the limitation of previous methods for Dynamic Programming by using multi-parametric programming techniques and avoids the need for global optimization at each stage of the Dynamic Programming procedure.
3. Algorithm for robust explicit/multi-parametric MPC in embedded control systems

A new algorithm for robust explicit/multi-parametric Model Predictive Control (MPC) for uncertain, linear discrete-time systems is proposed based on Dynamic Programming (DP), multi-parametric Programming and Robust Optimization. The presented algorithm features, i) a DP reformulations of the MPC optimization problem, ii) a robust reformulation of the constraints, and iii) a multi-parametric programming step, where the control variables are obtained as explicit functions of the state variable, such that the state and input constraints are satisfied for all admissible values of the uncertainty. A key feature of the proposed procedure, is that, as opposed to previous methods, it only solves a convex multi-parametric programming problem for each stage of the DP procedure.

3.1. Introduction

Model Predictive Control (MPC) is a control method which is well-known for its ability to handle multi-variable, constrained systems. In a typical MPC framework, the sequence of optimal control actions/inputs to the system are determined based on the prediction of the behaviour of the system over a finite time horizon (Rawlings and Mayne, 2009). This is achieved by solving repetitively an online constrained optimization problem, at each sampling instant during which the system state/output measurements (or estimates) become available. Only the first input is then applied to the system and the optimization is repeated at the next sampling time when the new set of measurements are available - an implicit feedback control policy is thus established. Despite the obvious benefits of MPC, its capabilities are
limited mainly due to the computational effort required for solving the online optimization problem of the MPC. This shortcoming can be overcome by using the so-called explicit/multi-parametric MPC (or mp–MPC) methods. In mp–MPC the online optimization problem involved in the MPC, is solved off–line using multi–parametric programming methods (Bank et al., 1982; Pistikopoulos and Dua, 2007b) to derive the control variables and the value function of the optimization problem as explicit functions of the system state variables, as well as the critical regions of the state–space where these functions are valid (Pistikopoulos and Dua, 2007a; Pistikopoulos, 2009) i.e. the governing control laws of the problem at hand are obtain. The key advantage of mp–MPC is that it can replace the online optimization problem of traditional MPC with a set of function evaluations, by directly applying the mp–MPC control laws to the system, and thus reducing the computational effort required for the MPC implementation.

The presence of model uncertainties(disturbances) in the process may significantly deteriorate the controller performance and most importantly cause constraints violations. Therefore, robust MPC methods have been developed which can explicitly handle the uncertainties and/or disturbances and guarantee the controller performance and satisfaction of constraints (Zafiriou, 1990; Bemporad and Morari, 1999; Kothare et al., 1996; Rawlings and Mayne, 2009). Robust MPC methods have been extensively studied in the open literature in the last decades and a significant number of publications exists (see (Rawlings and Mayne, 2009; Bertsekas and Rhodes, 1971; Langson et al., 2004, and references within)). Nevertheless, despite the benefits and popularity of both robust MPC and mp–MPC, the field of robust explicit/multi–parametric MPC has not yet received equal attention (see Table 3.1, due to the more complex nature of the optimization problem involved in MPC. Since the uncertainties and disturbances in the system appear in the right-hand (RHS) and left-hand (LHS) sides of the optimization model of the MPC, it is difficult to apply the current methods for multi–parametric programming and mp–MPC for the design of robust explicit controllers. With most of the current methods either focusing on solving the nominal mp–MPC problem, where no uncertainties are considered, or focusing on online robust MPC solutions, there are many issues and areas of robust explicit/multi–parametric MPC that are still unexplored (Pistikopoulos, 2009).
Table 3.1.: Key publications in robust explicit/multi–parametric MPC (mp–MPC)

<table>
<thead>
<tr>
<th>Focus Areas</th>
<th>Publications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust mp–MPC of linear systems with additive disturbances</td>
<td>Sakizlis et al. (2004); Alamo et al. (2005); Olaru and Ayerbe (2006); Raković et al. (2008)</td>
</tr>
<tr>
<td>Robust mp–MPC of linear systems with model uncertainties</td>
<td>Bemporad et al. (2003); Kouramas et al. (2009)</td>
</tr>
<tr>
<td>Min–max robust mp–MPC</td>
<td>Bemporad et al. (2003); Kerrigan and Maciejowski (2004); Alamo et al. (2005); Muñoz de la Peña et al. (2006)</td>
</tr>
<tr>
<td>Robust mp–MPC with linear input/output models</td>
<td>Muñoz de la Peña et al. (2005); Olaru and Ayerbe (2006)</td>
</tr>
</tbody>
</table>

In this chapter a robust explicit/multi–parametric MPC and an indicative list of publications is given in Table 3.1. In Sakizlis et al. (2004); Olaru and Ayerbe (2006) methods were developed for the linear quadratic regulation problem of linear system with additive disturbances and linear inequality constraints. In Sakizlis et al. (2004); Pannochia et al. (2005) the robust reference tracking problem was investigated and methods were developed for the design of robust tracking mp–MPC controllers. Min–max robust mp–MPC methods for linear systems of additive disturbances and inequality constraints were also developed in Alamo et al. (2005); Muñoz de la Peña et al. (2006). Finally, the case of linear uncertain systems was investigated in Bemporad et al. (2003); Muñoz de la Peña et al. (2004) and methods were developed for the design of robust mp–MPC controllers based on the use of linear objective functions (infinity or 1–norm based objective functions). Despite this important effort for solving certain classes of robust explicit/multi–parametric MPC, there are still some unresolved issues. The case of robust explicit/multi–parametric MPC for linear systems with model uncertainties with quadratic objective functions in the MPC optimization problem, has not yet been fully investigated. An other issue arises from the use of dynamic programming (DP) methods in the proposed procedures for solving the robust mp–MPC problem (Bemporad et al., 2003; Muñoz de la Peña et al., 2004). By applying DP methods, the MPC optimization problem is recast as a multi–stage optimization problem, which is then reduced
to a set of stage optimization sub–problems of smaller dimensionality (Bellman, 2003; Bertsekas, 2005). As it was shown in Faísca et al. (2008), solving each of the sub–problems of the DP with multi–parametric programming, results in a non–linear stage optimization problem, which requires the use of global optimization methods, even for the case of nominal mp–MPC.

This work presents a novel framework for the design of robust explicit/multi–parametric MPC controllers for linear dynamic systems with model uncertainties. The objective is to derive the optimal control laws (or policies) for the robust explicit/multi–parametric MPC problem, for which the state and input constraints are satisfied for all admissible values of the model uncertainty. Based on the previous work of Faísca et al. (2008), an algorithm is presented based on dynamic programming, multi–parametric programming and robust optimization (Ben-Tal and Nemirovski, 2000) that obtains the control variables as explicit functions of the system states.

3.2. Problem Definition

It is considered linear, discrete–time dynamic systems

\[ x_{t+1} = f(x_t, u_t) := Ax_t + Bu_t \]  \hspace{1cm} (3.1)

where \( x_t \in \mathbb{R}^n \) is the state vector, \( u_t \in \mathbb{R}^m \) is the vector of system inputs and the system matrices \( A, B \) are uncertain in that they are given as

\[ A = A_0 + \Delta A, \quad B = B_0 + \Delta B \]  \hspace{1cm} (3.2)

\[ \Delta A \in A = \{ \Delta A \in \mathbb{R}^{n \times n} | -\epsilon_a|A_0| \leq \Delta A \leq \epsilon_a|A_0| \} \]  \hspace{1cm} (3.3)

\[ \Delta B \in B = \{ \Delta B \in \mathbb{R}^{n \times n} | -\epsilon_b|B_0| \leq \Delta B \leq \epsilon_b|B_0| \} \]  \hspace{1cm} (3.4)

where \( A_0, B_0 \) are their nominal values and \( \Delta A, \Delta B \) are unknown but bounded matrices with \( \epsilon_a, \epsilon_b \in \mathbb{R}^{n \times n} \). The system states and inputs are subject to the following linear inequality constraints

\[ \mathcal{X} = \{ x \in \mathbb{R}^n | Gx \leq w \} \]  \hspace{1cm} (3.5)

\[ \mathcal{U} = \{ u \in \mathbb{R}^m | Mu \leq \mu \} \]  \hspace{1cm} (3.6)

where \( G \in \mathbb{R}^{n_g \times n}, \ w \in \mathbb{R}^{n_g}, \ M \in \mathbb{R}^{m_M \times m}, \mu \in \mathbb{R}^{m_M} \).
Assumption 1. It is assumed that $X, U$ are convex, compact polyhedral sets given by (3.5)–(3.6) and contain the origin in their interior, i.e. $0 \in X, U$.

Remark 12. The type of system uncertainty described in (3.2), is used mainly to describe systems for which the nominal values of the system matrices $A_0, B_0$ have been obtained by system model approximations (such as linearization or model identification), while $\Delta A, \Delta B$ describe the approximation error.

It is defined the Robust Explicit/multi-parametric Model Predictive Control (MPC) problem as follows

$$V_N^*(x) = \min_{U \in \mathcal{U}} J(U, x)$$

where $U \in \mathcal{U}_N \subset \mathbb{R}^{N_\mu}$ is the control sequence of the current and future control inputs and $N$ is the prediction horizon and $\mathcal{T}$ is the terminal constraint set given by

$$\mathcal{T} = \{x \in \mathbb{R}^n \mid Tx \leq \tau\}$$

where $T \in \mathbb{R}^{nT \times n}$, $\tau \in \mathbb{R}^{nT}$.

Assumption 2. The matrices $Q, P$ are positive semi–definite, $R$ is positive definite and $\mathcal{T}$ contains the origin in its interior.

The objective of this work is to obtain a control sequence $U$ and in extension the optimal control variables $u_t$ for the optimization (3.7) as functions of the state variables $x_t$, such that the state, input and terminal constraints are satisfied for all possible values of the uncertain system matrices $\Delta A, \Delta B$. It is called such as solution for the explicit/multi-parametric MPC problem a robust solution or robust explicit/multi-parametric control law.
**Definition 1.** A control sequence $U^* = \{u^*_0, u^*_1, \ldots, u^*_{N-1}\}$ is a robust solution for the explicit/multi-parametric MPC problem (3.7) if it satisfies the state and input constraints (3.9), (3.10) and (3.11) for all admissible values of $\Delta A \in A$, $\Delta B \in B$.

It is obvious from (3.7), that deriving the control variables $u_t$ as explicit functions of the states $x_t$ is not possible with the current nominal mp–MPC methods Bemporad et al. (2002); Pistikopoulos and Morari (2002b), due to the presence of the uncertain matrices $\Delta A$, $\delta B$. A method was presented in Kouramas et al. (2009) for solving (3.7) based on robust optimization and multi–parametric programming methods, which however results in a large, complex robust multi–parametric Quadratic Programming problem.

**Remark 13.** Problem (3.7) was also investigated in Bemporad et al. (2003); Muñoz de la Peña et al. (2006) for linear objective functions. A method was proposed based on dynamic programming and min–max linear programming (LP), in which the linear objective is substituted by a slack variable and is added as an additional linear constraint in the optimization problem, thus leading to a multi–parametric linear program. However, this method is not directly applicable for problem (3.7) in which the objective function is quadratic and hence, it would have resulted in an optimization problem with quadratic constraints. Additionally the use of dynamic programming in Bemporad et al. (2003) results in the generation of overlapping critical regions: in order to find the solution in the intersection of the critical regions a comparison optimization problem has to be solved which is in general a global optimization problem.

In this work it is described an algorithm for solving (3.7) based on dynamic programming, multi–parametric programming and robust optimization methods, which avoids the need for global optimization and only solves a convex multi–parametric programming problem for each stage of the dynamic programming procedure.

### 3.2.1. Robust multi-parametric programming

Consider the following class of robust Linear Programming (LP) problems Ben-Tal and Nemirovski (2000)

$$\min_u c^T u \quad \text{s.t.} \quad Au \leq b, \quad u_{lo} \leq u \leq u_{up}$$

(3.14)
where matrix \( \mathbf{A} \) is uncertain in that its true value is not known but is given as \( \mathbf{A} = \mathbf{A}_0 + \Delta \mathbf{A} \) where \( \mathbf{A}_0 \) is its nominal value and \( \Delta \mathbf{A} \) is only known to be bounded in \( |\Delta \mathbf{A}| \leq \epsilon \mathbf{A}_0 (\epsilon \in \mathbb{R}^{n \times n}) \). The objective is to obtain an optimal solution for (3.14) that is immune against the uncertainty i.e. which satisfies the linear inequality constraints for all possible values of \( |\Delta \mathbf{A}| \leq \epsilon \mathbf{A}_0 \). Following the robust LP method of Ben-Tal and Nemirovski (2000), a robust solution for the robust LP (3.14) can be obtained by solving the so called robust counterpart (RC) problem of Ben-Tal and Nemirovski (2000)

\[
\min_u c^T u \quad \text{s.t.} \quad \mathbf{A}_0 u \leq b, \quad \mathbf{A}_0 u + \epsilon |\mathbf{A}_0||u| \leq b + \delta \max[1,|b|] \quad (3.15)
\]

where \( \delta \) is a given infeasibility tolerance, \( 1 \) is a vector of 1 of the same dimension as \( b \) and max is taken element-wise i.e. \( \max[1,|b|]=\max[1,b_i] \). The RC problem (3.14) is a nonlinear problem, however it is equivalent to the Interval Robust counterpart (IRC) problem given below Ben-Tal and Nemirovski (2000)

\[
\min_u c^T u \quad \text{s.t.} \quad \mathbf{A}_0 u \leq b, \quad \mathbf{A}_0 u + \epsilon |\mathbf{A}_0|y \leq b + \delta \max[1,|b|] \\
\quad \quad -y \leq u \leq y, \quad u_{lo} \leq u \leq u_{up}
\]

which is a LP problem.

The IRC problem obtains a robust solution \( u^* \) for the robust LP (3.14) i.e. a solution which is both feasible and immune against the uncertainty for (3.14) Ben-Tal and Nemirovski (2000).

**Remark 14.** A feasible solution \((u, y)\) for (3.16) satisfies the following inequality \( \mathbf{A}_0 u + \Delta \mathbf{A} u \leq \mathbf{A}_0 u + \epsilon \mathbf{A}_0||u| \leq \mathbf{A}_0 u + \epsilon \mathbf{A}_0|y \leq b + \delta \max[1,|b|] \), hence it is immune against the uncertainty.

**Remark 15.** Note that the above method can be easily extended for the case of robust convex Quadratic Programming problems where the objective function is given as a convex quadratic function of \( u \), \( \min_u u^T Qu, \quad Q > 0 \).

The above method was extended in Kouramas et al. (2009) for the case
of multi–parametric Quadratic Programming problems

\[
\min_u u^T Qu \\
\text{s.t. } Au \leq b + Fx, \quad u_{lo} \leq u \leq u_{up}, \quad x_{lo} \leq x \leq x_{up}
\]  

(3.17)

where \( u \) is the optimization variable, \( x \) is the vector of parameters and \( A, F \) are uncertain in that they are given by \( A = A_0 + \Delta A, |\Delta A| \leq \epsilon_A |A_0| \) and \( F = F_0 + \Delta F, |\Delta F| \leq \epsilon_F |F_0| \), where \( A_0, F_0 \) are the nominal values of \( A, F \) and \( \Delta A, \Delta F \) are their uncertain parts given as percentages of the nominal values. The objective here is to obtain an explicit solution \( u(x) \) which is both feasible and immune against the uncertainty. The Robust Counterpart problem for (3.17) is given by

\[
\min_u u^T Qu \\
\text{s.t. } A_0 u \leq b + F_0 x \\
A_0 u + \epsilon_A |A_0||u| \leq b + F_0 x - \epsilon_F |F_0||x| + \delta \max[1, |b|] \\
u_{lo} \leq u \leq u_{up}, \quad x_{lo} \leq x \leq x_{up}
\]  

(3.18)

The IRC equivalent for (3.19) is then given as follows Kouramas et al. (2009)

\[
\min_u u^T Qu \\
\text{s.t. } A_0 u \leq b + F_0 x \\
A_0 u + \epsilon_A |A_0||u| \leq b + F_0 x - \epsilon_F |F_0||x| + \delta \max[1, |b|] \\
y - y \leq u \leq y, \quad -z \leq x \leq z \\
u_{lo} \leq u \leq u_{up}, \quad x_{lo} \leq x \leq x_{up}
\]  

(3.19)

which is a convex mp-QP problem Pistikopoulos and Morari (2002b). The solution of the IRC (3.19) is an explicit Piecewise Affine (PWA) function \( u(x) \) of the parameters \( x \) Pistikopoulos and Dua (2007b).

\textbf{Lemma 2.} A feasible solution \( u(x) \) for (3.19) is a robust solution for (3.17).

\textbf{Proof} If \( u(x) \) is a feasible solution of (3.19) then the following inequality
is satisfied

\[ A_0 u + \Delta A u \leq A_0 u + \epsilon_A |A_0| |u| \leq A_0 u + \epsilon_A |A_0| y \]
\[ \leq b + F_0 x - \epsilon_f |F_0| z + \delta \max[|I|, |b|] \]
\[ \leq b + F_0 x - \epsilon_f |F_0| |x| + \delta \max[|I|, |b|] \]
\[ \leq b + F_0 x + \Delta F x + \delta \max[|I|, |b|] \]

Therefore the inequality constraints of (3.17) are satisfied for all values of \( \Delta A, \Delta F \) and \( u(x) \) is a robust solution for (3.17).

**Remark 16.** Note that for control problems, such as MPC, involving hard state and inputs constraints, no infeasibility tolerance can be allowed and hence \( \delta = 0 \).

### 3.3. Algorithm for explicit/multi-parametric MPC

The proposed method for the solution of the robust explicit/multi-parametric MPC problem is realized in three main steps

1. a dynamic programming step: the MPC optimization (3.7) is recast in a multi-stage optimization setting and solved as a set of single stage optimizations
2. a robust reformulation of the constraints: constraints are reformulated at each stage to account for the worst-effect of the uncertainty
3. a multi-parametric programming step: the reformulated stage optimizations are solved as convex multi-parametric programming problems

These steps are now described in more detail in the following sections.

#### 3.3.1. Dynamic Programming step - Multi-stage Optimization

Following the main idea of dynamic programming and multi-stage optimization presented in chapter 2, (3.7) is expressed as a multi-stage optimization problem where the time instant \( t \) represents each of the stages of the problem. This multi-stage problem (3.7) then can be decomposed into a set of
smaller, stage problems as given in the following expression

\[ V_t(x_t) = \min_{u_t \in \mathcal{U}} J_t(u_t, x_t) \]

\[ = \min_{u_t \in \mathcal{U}} \sum_{i=t}^{N-1} \left\{ x_i^T Q x_i + u_i^T R u_i \right\} + x_N^T P x_N \]  \quad (3.20)

s.t.  \( x_{i+1} = A x_i + B u_i \)
\( x_t \in \mathcal{X}, \ u_t \in \mathcal{U} \)
\( x_{t+1} \in \mathcal{X}_{t+1} \)

The optimization problem (3.20) is solved stage-wise, starting from \( t = N - 1 \) and solving it repetitively backwards until \( t = 0 \), where the only optimization variable is the control variable \( u_t \) at the current stage \( t \) and only the state and input constraints at the current stage are considered, given the optimal controls \( u_{t+1} = \mu_{t+1}(x_{t+1}), \ldots, u_{N-1} = \mu_{N-1}(x_{N-1}) \) from the solution of (3.20) at the previous stages. The additional constraint set \( x_{t+1} \in \mathcal{X}_{t+1} \) is the \( t + 1 \)-robust controllability set to the target set \( \mathcal{T} \) and is used to ensure that a feasible control \( u_t \) exists at the next stage \( t + 1 \) as well Rawlings and Mayne (2009)

\[ \mathcal{X}_t = \{ x \in \mathcal{X} \mid \exists u \in \mathcal{U} \ s.t. \ Ax + Bu \in \mathcal{X}_{t+1}, \ \forall \Delta A \in \mathcal{A}, \Delta B \in \mathcal{B} \}, \ \mathcal{X}_{N-1} = \mathcal{T} \]  \quad (3.21)

\( t \in [0, N - 1] \)

**Lemma 3** (Rawlings and Mayne (2009)). For all \( x \in \mathcal{X}_t \) there exists a feasible control sequence \( u_t, \ldots, u_{N-1} \) such that \( x_i \in \mathcal{X}_i \subset \mathcal{X} \) and \( u_i \in \mathcal{U} \) for \( i \geq t \) and \( x_{N-1} \in \mathcal{T} \), for all \( \Delta A \in \mathcal{A}, \Delta B \in \mathcal{B} \).

By including the constraint \( x_{t+1} \in \mathcal{X}_{t+1} \) in (3.20) it is ensured that the state \( x_{t+1} \) in the next time instant will lie in the set of states for which a feasible control (satisfying all constraints for all possible values of the uncertainty) \( u_{t+1} \) exists for the problem (3.20) at the next stage \( t + 1 \). Therefore, a feasible control is ensured for both stage \( t \) and \( t + 1 \). It will call \( x_{t+1} \in \mathcal{X}_{t+1} \) as the feasibility constraints. Methods for the calculation of \( \mathcal{X}_t \) will be discussed later in this chapter in Section 3.3.1. Since \( \mathcal{X}, \mathcal{U} \) and \( \mathcal{T} \) are polyhedral sets with the origin in their interior, the \( t \)-robust controllability set is also a polyhedral set containing the origin in its interior (see Rawlings

57
and Mayne (2009) and also Section 3.3.1)

\[ X_t = \{ x \in X \mid H^t x \leq h^t \} \] \hspace{1cm} (3.22)

**Remark 17.** Note that problem (3.20) is convex with respect to \( x_t, u_t, \ldots, u_{N-1} \) since the objective is a convex quadratic function and the constraints are linear.

**Remark 18.** Despite (3.20) being a convex problem, when traditional DP methods are applied, the solutions \( u_{t+1} = \mu_{t+1}(x_{t+1}), \ldots, u_{N-1} = \mu_{N-1}(x_{N-1}) \) of the previous stages are substituted in (3.20) to form a new problem on \( u_t \) and \( x_t \) only. Since each of \( u_{t+1} = f_{t+1}(x_{t+1}), \ldots, u_{N-1} = \mu_{N-1}(x_{N-1}) \) are piecewise affine (PWA) functions of the states, (3.20) becomes a nonlinear optimization problem at every stage, since the objective function is piecewise quadratic and the constraints are piecewise affine functions of \( x_t, u_t \). Global or combinatorial optimization method have to be used to solve (3.20) even for the nominal case (see Remark 2 and Muñoz de la Peña et al. (2004); Faísca et al. (2008)).

**Remark 19.** The complexity of (3.20) is also increased due to the presence of the uncertain matrices \( \Delta A, \Delta B \) in the objective function and the inequality constraints.

In view of remarks 17-19, the convexity of (3.20) can be used to derive a convex multi-parametric programming problem as shown next. First, by

1. considering \( u_t \) as the optimization variable for (3.20)
2. considering \( \theta_t = [x_t \, u_{t+1} \ldots u_{N-1}] \) as the vector of parameters for (3.20)
3. considering that the objective function only penalizes the nominal system \( (A = A_0, B = B_0) \) and thus substituting \( x_i = A^{i-t} x_t + \sum_{j=t}^{i-1} A^{i-1-j} B u_j \) for \( i = t + 1, \ldots, N \) in the objective, and
4. replacing \( x_{t+1} = A x_t + B u_t \) in the inequality constraints where \( A, B \) are given from (3.2)

58
we obtain the following reformulation of (3.20)

\[
V_t(x_t) = \min_{u_t \in U} \sum_{i=t}^{N-1} \left\{ \frac{1}{2} u_t^T H u_t + \theta_t^T F u_t \right\} + \theta_t Y \theta_t
\]

s.t. \[ H^T A x_t + H^T B u_t \leq h^t \]
\[ G x_t \leq w , \ M u_t \leq \mu \]  

Problem (3.23) is a robust multi-parametric Quadratic Programming problem Kouramas et al. (2009) since

- the objective function is a convex quadratic function of both the control variable \( u_t \) and the parameter vector \( \theta_t \),
- the inequality constraints are linear but uncertain due to the uncertain matrices \( A, B \).

**Remark 20.** It is a frequent practice taking the objective in (3.20) to penalize only the nominal system dynamics (see for the example the case of predictive tubes Rawlings and Mayne (2009)). Including the uncertainty in \( A, B \) in the objective function of (3.20) would have resulted in a complex minmax optimization problem requiring combinatorial and nonlinear optimization methods Wang and Rawlings (2004).

**Robust Reformulation step**

Since (3.23) is a robust mp-QP problem, its robust counterpart can be obtained from (3.23) as follows

\[
V_t(x_t) = \min_{u_t \in U} \sum_{i=t}^{N-1} \left\{ \frac{1}{2} u_t^T H u_t + \theta_t^T F u_t \right\} + \theta_t Y \theta_t
\]

s.t. \[ H^T A x_t + H^T B u_t \leq h^t \]
\[ H^T B_0 u_t + \epsilon_\beta |H^T B_0||u_t| \leq h^t - H^T A_0 x_t \]  
\[ - \epsilon_\alpha |H^T A_0||x_t| + \delta \max(1,|b|) \]
\[ G x_t \leq w , \ M u_t \leq \mu \]  

**Remark 21.** Inequality (3.25) ensures (see Lemma 2) that the control obtained by solving (3.24) is a robust solution for (3.20) (i.e. is feasible and satisfies the constraints for all values of \( \Delta A, \Delta B \)).
The RC problem (3.24) is a nonlinear problem due to the nonlinear inequality (3.25). However, this issue is resolved by considering the Interval Robust Counterpart (IRC) of (3.24). More specifically, a convex mp-QP can then be formed by taking the IRC equivalent problem of (3.24) (see (3.19)) as follows

\[
V_t(x_t) = \min_{u_t \in U} \sum_{i=1}^{N-1} \left\{ \frac{1}{2} u_t^T H u_t + \theta_t^T F u_t \right\} + \theta_t Y \theta_t \quad (3.26)
\]

s.t. \( H^t A x_t + H^t B u_t \leq h_t \)
\[
H^t B_0 u_t + \epsilon \beta \|H^t\| B_0 y_t \leq h - H^t A_0 x_t
\]
\[
- \epsilon \alpha \|H^t\| A_0 z_t + \delta \max[1, |b|]
\]
\[
- y_t \leq u_t \leq y_t, -z_t \leq x_t \leq z_t
\]
\[
G x_t \leq w, \ M u_t \leq \mu
\]

Multi–parametric Programming

Since (3.26) is a convex multi–parametric Quadratic Programming problem, its optimal solution \( u_t \) is an explicit PWA function of \( \theta_t \) Pistikopoulos and Dua (2007b) which is given as follows

\[
u_t = f_t(\theta_t) = \left\{ K_i^t \theta_t + c_i^t \right\} \text{ if } \theta_t \in CR_i^t, \ i = 1, \ldots, s_t
\]

where \( CR_i^t = \left\{ \theta_t \mid \Pi_i^t \theta_t \leq \pi_i^t \right\} \) (3.28)

**Remark 22.** In the general DP setting, the value of \( u_t \) is derived at each stage \( t \) of the DP procedure based on the values of \( x_t \) and of the future control decisions \( u_i, \ i \geq t + 1 \) that are derived at the previous stages by means of the cost-to-go function: the cost-to-go function is directly obtained by substituting the control variables of the previous stages in the objective function Bertsekas (2005).

**Remark 23.** In view of the above remark, expression (3.28) is the functional relation between the control decision at the current stage and the state \( x_t \) and future control decisions.
Parameter reduction of the explicit solution

Equation (3.28) expresses \( u_t \) as an explicit function of \( \theta_t \). However, the objective is to obtain \( u_t \) as only an explicit function of the current state \( x_t \). In order to do so, the future control variables \( u_i, i \geq t + 1 \) have to be eliminated from (3.28), thus reducing the vector of parameters only to \( x_t \). A method for the reduction of the parameters in (3.28) was presented in Faisca et al. (2008), which is briefly overviewed in the appendix. The method relies on i) substituting the solutions \( u_i = \mu_i(x_i) \) of the previous stages \( i \geq t + 1 \) into (3.28) and ii) performing linear algebraic manipulations to eliminate the \( u_i, i \geq t + 1 \) from the extended solution (3.28) and derive \( u_t \) as only an explicit function of \( x_t \) (see Appendix)

\[
 u_t = \mu_t(x_t) = \{F_t^ix_t + f_t^i \mid x_t \in \mathcal{CR}_t^1, \; i = 1, \ldots, q_t \} 
\]

where \( \mathcal{CR}_t^i = \{x \in \mathbb{R}^n \mid A_t^ix \leq b_t^i \} \) (3.29)

Remark 24. Note that with this methodology (see Appendix), the convex multi-parametric programming problem (3.26) is solved first and then the solutions \( u_i = \mu_i(x_i) \), \( i \geq t + 1 \) of the previous stages are substituted in (3.28) to derive (3.29). This allows us to treat (3.26) as a convex multi-parametric quadratic program. In the traditional DP procedure, the solutions \( u_i = \mu_i(x_i), i \geq t + 1 \) are first substituted to (3.26) which then becomes a nonlinear problem Faisca et al. (2008).

Feasibility constraint

The \( t \)-robust controllability set for the feasibility constraint in (3.20) can be obtained either

1. from (3.21) by performing set theoretic computations, (see Rawlings and Mayne (2009) and references within), or

2. by

\[
 X_t = \bigcup_{i=1}^{q_t} \mathcal{CR}_t^i 
\]

(3.30)

i.e. as the set of all feasible states \( x_t \) for which (3.29) is optimal for (3.20).
3.3.2. Algorithm for explicit/multi-parametric MPC

An algorithm for the solution of the explicit/multi-parametric MPC problem can now be proposed, based on the procedure described in 3.3.1–3.3.1. Algorithm 2 describes the main steps of the proposed algorithm. The algorithm starts at stage $t = N - 1$, and iterates through Steps 2-6 of Algorithm 2 for each stages $t$, until $t = 0$. In Step 3 the mp-QP problem (3.26) is solved for the current stage $t$ to derive $u_t = f_t(\theta_t)$ as a function of the state and future control variables. Then, in Step 4 the elimination procedure described in Appendix 1 is applied to obtain the explicit control $u_t = \mu_t(x_t)$. After the algorithm terminates at the last stage $t = 0$, a sequence of explicit feedback control laws $U = \{\mu_0(x_0), \ldots, \mu_{N-1}(x_{N-1})\}$ is computed, where each control law $\mu_t(x_t)$ is given by a PWA expression (3.29).

The main property of the Algorithm is given in the following proposition

**Proposition 1** The control sequence $U = \{u_0, \ldots, u_{N-1}\}$, where $u_t = \mu_t(x_t)$, $t = 0, \ldots, N-1$, is a robust solution for the explicit/multi-parametric MPC problem (3.7).

**Proof** Given Lemma (2), the solution $u_t = \mu_t(x_t)$ (3.29) of the mp-QP problem (3.26), obtained at each step of algorithm (2) is a robust solution of (3.20) i.e. satisfies the inequality constraints of (3.20) for all values of the uncertain matrices $\Delta A, \Delta B$. In addition, from Lemma (3), the constraint $x_{t+1} \in X_{t+1}$ in (3.20) ensures that a control sequence $\{u_t, \ldots, u_{N-1}\}$ exists which satisfies the constraints $x_i \in X$ and $u_i \in U$ for all values of $\Delta A, \Delta B$. Therefore, $U$ is a robust solution for (3.7) since it satisfies the state and input constraints of (3.7). i.e. is feasible and immune for (3.7) (see Definition 1).
Algorithm 2 Algorithm for robust explicit/multi–parametric MPC

Data: $A, B, Q, R, P, N, \mathcal{X}, \mathcal{U}, T, \epsilon_a, \epsilon_b$

Output: A control sequence of explicit control laws $\{u_t(x_t), \ldots, u_{N-1}(x_{N-1})\}$ for the robust explicit/multi–parametric MPC problem

1: Set $t = N - 1$ (start at stage $N - 1$)
2: Repeat for each stage $t$
3: Solve (3.26) with $u_t$ being the optimization variable and $\theta_t = \{x_t, u_{t+1}, \ldots, u_{N-1}\}$ being the parameter vector. Obtain $u_t = f_t(\theta_t)$ from (3.28).
4: Eliminate $u_{t+1}, \ldots, u_{N-1}$ to obtain $u_t = \mu_t(x_t)$ (see parameter reduction method Appendix).
5: Obtain $\mathcal{X}_t = \bigcup_{i=1}^{\mathcal{R}_t} \mathcal{R}_t$.
6: Set $t = t - 1$
7: Until final stage $t = 0$ has been reached

The main advantages and characteristics of the proposed procedure are now summarized below:

- In the proposed procedure, the explicit/multi–parametric MPC problem (3.7) is recast as a multi-stage optimization problem by applying DP methods, thus avoiding the solution of a large robust multi–parametric programming problem with a demanding computational cost as in Kouramas et al. (2009).

- Each stage (3.20) optimization problem of the proposed DP procedure, is solved only as a convex multi–parametric quadratic programming problem (3.26). The explicit solution (3.29) to each of the stage problems (3.20) is then derived by performing linear algebraic operations on the solution (3.28).

- Robustification of the constraints of (3.20) against the uncertainty, are used to ensure that the control solution (3.29) is a robust solution of (3.20) and hence the state and input constraints for the incumbent stage are satisfied for all values of the uncertainty.
• The proposed algorithm obtains a robust solution for the explicit/multi-parametric MPC problem (Proposition 1) i.e. the resulting control functions satisfy the system (input and output) constraints for all possible values of the uncertain matrices $\Delta A, \Delta B$.

• The parametric programming techniques have computational limitations on the size of the problem that can be solved, due to the generation of huge matrices. A typical problem of 8 states, 4 inputs, 4 outputs and horizon 15 is the maximum we have solved in MATLAB.

**Controller implementation**

Since problem (3.7) is an explicit/multi-parametric MPC problem, only the first control move of the control sequence $U$ is applied to the system

$$u(x) = \mu_0(x)$$

where $\mu_0(x)$ is a piecewise affine feedback control law. A feedback control policy is thus established. A different approach will be to implement the whole control sequence $U$, by applying each $u_t = \mu_t(x_t)$ at each time for the given state $x_t$, although this will results in the loss of the MPC’s prediction and feedback properties.

**Computational complexity**

The number of linear inequalities for problem (3.26) is $\eta_{ineq} = m_M + 2n_{\mathcal{X}_t}$ where $n_{\mathcal{X}_t}$ is the number of inequalities for the set $\mathcal{X}_t$ (3.22). An upper bound for the number of critical regions for (3.26) can then be obtained as follows (see Pistikopoulos and Morari (2002b); Pistikopoulos and Dua (2007a))

$$\eta_{DP} \leq \sum_{k=0}^{\eta-1} k!(m_M + 2n_{\mathcal{X}_t})^k$$

where $\eta = (m_M + 2n_{\mathcal{X}_t})!/(m_M + 2n_{\mathcal{X}_t} - i)!i!$.

The key issues regarding the computational complexity of Algorithm 1 are then discussed as follows.
• Algorithm 1 solves a set of convex mp-QP problems (3.26) in which the number of linear inequality constraints is determined by the number of input and feasibility constraints only.

• The number of inequalities \( n_X \) only depends, as it can be seen from (3.21), on the dimensions of the system matrices \( A, B \), the uncertainty set (3.3), (3.4) and the state and input constraints \( X, U \) and it is independent of the prediction horizon \( N \).

• The prediction horizon of (3.7) does not affect \( \eta_{\text{ineq}} \) and hence the number of critical regions \( \eta_{\text{DP}} \). This is the case when (3.7) is solved as a single mp-QP problem, even for nominal case, and which is an important factor for the increase of the complexity of the problem.

On the other hand, the number of parameters at each stage increases since the number of future inputs considered in each stage increases. Hence, the number of parameters that have to be eliminated in (3.28) increases, in order to derive the explicit solution (3.29) and thus affecting the computational complexity of the algebraic computations that are needed to derive the explicit control (3.29). Nevertheless, the operations performed at this step of the proposed algorithm rely only on linear operations such as Fourier-Motzkin or Gauss elimination.

3.4. Examples

3.4.1. Example 1

The proposed methodology is now demonstrated with the following illustrative example

\[
A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \epsilon_{\alpha} = \epsilon_{\beta} = 0.2
\]

\[X = x \in \mathbb{R}^n \mid -x_{\min} \leq x \leq x_{\max}, \quad x_{\min} = x_{\max} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}\]

\[U = u \in \mathbb{R} \mid -1 \leq u \leq 1\]
\[ Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = 1, \quad P = \begin{bmatrix} 2.6005 & 2.081 \\ 2.081 & 3.3306 \end{bmatrix}, \quad N = 3 \]

The target set is assumed to be equal to \( \mathcal{T} = \mathcal{X} \) and \( \delta = 0 \). Algorithm 1 is applied to the above problem and each of its steps are described below. The algorithm starts at stage \( N - 1 = 2 \)

**Stage 2**

In step 3 of Algorithm 2, it is considered the following stage optimization (3.20)

\[
V_2(x_2) = \min_{u_2 \in \mathcal{U}} x_2^T Q x_2 + u_2^T R u_2 + x_3^T P x_3 \\
\text{s.t. } x_3 = A x_2 + B u_2 \\
\quad - x_{\text{min}} \leq x_3 \leq x_{\text{max}} \\
\quad -1 \leq u_2 \leq 1
\]

Therefore, only the constraints and the optimization variable \( u_2 \) are considered at this stage. The above optimization can then be solved as an mp-QP problem, as it was shown in (3.26), with \( u_2 \) being the optimization variable and \( x_2 \) the parameters. The mp-QP method of Pistikopoulos and Morari (2002b) is then used to derive the explicit solution \( u_2 = \mu_2(x_2) \), which is given in Table 3.2. Since the control in this stage stage is already an explicit function of the state \( x_2 \), the parameter reduction in step 5 of Algorithm 2 is not necessary. In step 6 the controllability set \( \mathcal{X}_2 \) is obtained by taking the union (3.30) of all critical regions in Table 3.2 which is given by

\[ \mathcal{X}_2 = \{ x \in \mathbb{R}^2 \mid H^2 x \leq h^2 \} \]
The critical regions of $u_2 = \mu_2(x_2)$ are also shown in Figure 3.1.

Table 3.2.: Explicit solution $u_2 = \mu_2^*(x_2)$ for Stage 2 of the Example.

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
</table>
| $u_2 = 1$   | $-x_1^2 - x_2^2 \leq 10$  
|             | $0.3845x_1^2 + x_2^2 \leq -0.7078$  
|             | $-x_2^2 \leq 10, -x_1^2 \leq 10, x_1^2 \leq 10$ |
| $u_2 = -0.5432x_2^1 - 1.4127x_2^2$ | $-0.3845x_1^2 - x_2^2 \leq 0.7078$  
|             | $-0.3845x_1^2 - x_2^2 \leq -0.0070$  
|             | $x_1^2 \leq 10, -x_1^2 \leq 10$ |
| $u_2 = -0.4805x_2^1 - 1.2496x_2^2$ | $-0.3845x_1^2 - x_2^2 \leq 0.0070$  
|             | $0.3845x_1^2 + x_2^2 \leq 0.0070$  
|             | $-x_1^2 \leq 10, x_1^2 \leq 10$ |
| $u_2 = -0.01$ | $-0.3845x_1^2 - x_2^2 \leq 0.0089$  
|             | $0.3845x_1^2 + x_2^2 \leq 0.0107$  
|             | $x_1^2 \leq 10, x_2^2 \leq 10$ |
| $u_2 = -0.3569x_2^1 - 0.9281x_2^2$ | $-0.3845x_1^2 - x_2^2 \leq -0.0107$  
|             | $0.3845x_1^2 + x_2^2 \leq 1.0774$  
|             | $-10 \leq x_1^2 \leq 10$ |
| $u_2 = -1$  | $-0.3845x_1^2 - x_2^2 \leq -1.0774$  
|             | $x_1^2 + 0.8181x_2^2 \leq 9.0909$  
|             | $0.8181x_1^2 + x_2^2 \leq 9.0909$  
|             | $x_1^2 + 0.9090x_2^2 \leq 9.0909$  
|             | $0.9090x_1^2 + x_2^2 \leq 9.0909$  
|             | $x_2^1 + x_2^2 \leq 9.0909$  
|             | $-10 \leq x_1^2 \leq 10$ |
Stage 1

In step 3 of Algorithm 2 for this stage of the example, the following stage optimization is considered (3.20)

\[
V_1(x_1) = \min_{u_1 \in U} x_1^T Q x_1 + u_1^T R u_1 + x_2^T Q x_2 + u_2^T R u_2 + x_3^T P x_3 \\
\text{s.t. } x_3 = A x_2 + B u_2 \\
x_2 = A x_1 + B u_1 \\
- x_{\text{min}} \leq x_1 \leq x_{\text{max}} \\
- x_{\text{min}} \leq x_2 \leq x_{\text{max}} \\
H^2 x_2 \leq h^2 \\
- 1 \leq u_1 \leq 1
\]

(3.32)

Again, in the above stage optimization problem, only the control variable \(u_1\) and the constraints at the current stage are considered, with the only difference that the feasibility constraint \(H^2 x_2 \leq h^2\) has been added to ensure that the state \(x_2\) at the next stage belongs to the set of feasible states of (3.31). Problem (3.33) can then be solved as an mp-QP problem, as shown in (3.26), with \(u_1\) being the optimization variable and \(x_1, u_2\) being the parameters. The explicit solution for (3.33) \(u_1 = f_1(x_1, u_2)\) (of \(u_1\) as a
function of \(x_1, u_2\) consists in total of 61 critical regions and three of these critical regions with their corresponding controls are shown in Table 3.4. In step 5 of Algorithm 2 the parameter reduction method (Appendix) is applied to obtain the \(u_1 = \mu_1(x_1)\) as a function of \(x_1\) only, by incorporating expressions in Table 3.2 in the expressions in Table 3.4 and then solving for \(u_1\). The explicit control law \(u_1 = \mu_1(x_1)\) now consists of 105 critical regions. Due to space limitations, it is presented three of these critical regions in Table 3.3. The critical regions of \(u_1 = \mu_1(x_1)\) are also presented graphically in Figure 3.2. The feasibility set \(X^1\) is then obtained in step 5 from the union (3.30), as it was shown in stage 2, but is not presented here due to space limitations.

![Critical regions of the solution](image)

Figure 3.2.: Critical regions of the solution \(u_1 = \mu_1^*(x_1)\) for stage 1 for example 1.

Stage 0

In the final stage of the problem the following stage optimization problem is considered
Table 3.3.: Explicit solution $u_1 = \mu_1^*(x_1)$ for Stage 1 of Example 1.

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
</table>
| $u_1 = 1$   | $0.2777x_1^1 + x_1^2 \leq -1.2335$  
|             | $-x_1^1 - x_1^2 \leq 10$           
|             | $-0.5x_1^1 - x_1^2 \leq 5.5$       
|             | $x_1^1 \leq -0.01, x_2^1 \leq -0.01$ |
| $u_1 = -4411x_1^1 - 1.3145x_1^2$ | $-0.3245x_1^1 - x_1^2 \leq 0.0085$  
|             | $x_1^1 \leq 0.005, x_2^1 \leq 0.005$ |
| $u_1 = -0.3903x_1^1 - 1.2028x_1^2 - 0.0002$ | $-0.3245x_1^1 - x_1^2 \leq 0.0085$  
|             | $x_1^1 \leq 0.005, x_2^1 \leq 0.005$ |

\[ V_1(x_1) = \min_{u_0 \in \mathcal{U}} x_0^T Q x_0 + u_0^T R u_0 + x_1^T Q x_1 + u_1^T R u_1 \]
\[ + x_2^T Q x_2 + u_2^T R u_2 + x_3^T P x_3 \]
\[ \text{s.t. } x_3 = A x_2 + B u_2 \]
\[ x_2 = A x_1 + B u_1 \]
\[ x_1 = A x_0 + B u_0 \]
\[ -x_{\min} \leq x_0 \leq x_{\max} \]
\[ -x_{\min} \leq x_1 \leq x_{\max} \]
\[ -x_{\min} \leq x_2 \leq x_{\max} \]
\[ H^1 x_1 \leq h^1 \]
\[ -1 \leq u_0 \leq 1 \]

Algorithm 2 is also applied for this stage in the same way as in the two previous steps. The explicit solution $u_0 = \mu_0(x_0)$ for (3.33) consists of 344 critical regions. Three of these critical regions are shown in Table 3.5 while all critical regions are shown in Figure 3.3. Note that since the controller will be applied in an MPC fashion this $u_0 = \mu_0(x_0)$ also the control action that will be applied to the system. Each of the mp-QP problems in the above stages of this example, where solved by using the method of Pistikopoulos and Morari (2002b) and the POP software for Matlab PAROS (2009). A simulation of the system trajectory for the implementation of the obtained
robust explicit MPC controller $u(x) = \mu_0(x_0)$ is given in Figure 3.3, where the uncertain matrices $A, B$ are assumed to take their values at the vertices of the sets (3.3), (3.4). Note that the trajectory of the system remains in the set of feasible states for (3.33) despite the effect of the uncertainty.

Figure 3.3: Critical regions of the solution $u_0 = \mu_0(x_0)$ for stage 0 for Example 1.

Finally, Table 6 presents the execution times and number of parameters for an indicative number of stages of algorithm 2, when the horizon for prob-
Table 3.5.: Explicit solution $u_0 = \mu_0(x_0)$ for Stage 0 of Example 1.

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_0 = 1$</td>
<td>$0.2173x_0^0 + x_0^1 \leq -1.748$</td>
</tr>
<tr>
<td></td>
<td>$-0.5x_0^1 - x_0^2 \leq 5.5$</td>
</tr>
<tr>
<td></td>
<td>$-0.3333x_0^0 - x_0^3 \leq 4.3333$</td>
</tr>
<tr>
<td></td>
<td>$x_0^1 \leq -0.01$, $x_0^2 \leq -101$</td>
</tr>
</tbody>
</table>

$u_0 = -0.3005x_0^1 - 1.2098x_0^0 - 0.8443$

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_0 = -0.3357x_0^1 + x_0^2 \leq -5.73$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$-0.2484x_0^0 - x_0^3 \leq 0.6896$</td>
</tr>
<tr>
<td></td>
<td>$x_0^1 \leq 10$</td>
</tr>
</tbody>
</table>

$u_1 = -0.4712x_0^1 - 1.3486x_0^2$

<table>
<thead>
<tr>
<th>Control law</th>
<th>Critical Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1 = -x_0^0 - 0.9041x_0^1 \leq -0.0789$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$x_0^1 + x_0^2 \leq -0.01$</td>
</tr>
<tr>
<td></td>
<td>$-0.3493x_0^0 - x_0^3 \leq 0.7414$</td>
</tr>
</tbody>
</table>

Table 3.6.: Execution times and number of parameters for each stage of Algorithm 2 – Example 1.

<table>
<thead>
<tr>
<th>Stages</th>
<th>10</th>
<th>9</th>
<th>5</th>
<th>3</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution time (sec)</td>
<td>0.22</td>
<td>0.46</td>
<td>2.78</td>
<td>4.63</td>
<td>6.21</td>
<td>8.10</td>
</tr>
<tr>
<td># of parameters</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Note that the number of parameters for each stage increases and hence the execution time for each stage increases (due to the increasing number of parameters that have to be eliminated in each stage). Additionally, since $u$, $y$ and $z$ are the new optimization variables in each stage, the number of optimization variables increases by $n + m = 3$. Nevertheless, the execution time for each stage is not significant, and the algorithm can be extended to larger horizons. It is also expected that the execution time will increase for large values of the system inputs and and constraints.
3.4.2. Example 2.

Finally the proposed algorithm is demonstrated for the following linear discrete–time system with three states and one input

\[
A_0 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \epsilon_\alpha = \epsilon_\beta = 0.5
\]

where it is assumed that the uncertainty is 50% of its nominal value, with the following state, input and terminal constraints

\[
x_{\min} = \begin{bmatrix} -2 \\ -2 \\ -2 \end{bmatrix} \leq x_k \leq \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix} = x_{\max}, \quad -2 \leq u_k \leq 2, \quad \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \leq x_N \leq \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}
\]

and

\[
Q = I_3, \quad R = 0.1, \quad P = I_3, \quad N = 2
\]

where \( I_3 \) is the \( 3 \times 3 \) identity matrix. Since \( N = 2 \), Algorithm 2 decomposes the above in two stages, stage 1 and stage 0. Six critical regions are obtained for stage 1, while 171 are obtained for stage 0. The critical regions for stage 1 and 0 are shown in Figures 3.4, 3.5 respectively. Note, that the number of critical regions in stage 0 has increased compared to the number of critical regions in stage 1. This is due to many different comparisons that are performed, when the solution of stage 1 is replaced in stage 2 to eliminate the future control variable \( u_1 \). A simulation of the state trajectory of the linear system with the robust explicit/multi–parametric MPC controller is shown in Figure 3.5, while the time profiles of the states and inputs of the system are given in 3.6. For this simulation, the values of the uncertain matrices \( \Delta A, \Delta B \) were varied between their extreme values.

3.5. Concluding Remarks

In this chapter, an algorithm for robust explicit/multi–parametric MPC was presented, that features three key steps based on DP, robust optimization and multi–parametric programming. The algorithm overcomes the need for global optimization when explicit/multi–parametric MPC is solved with
Figure 3.4.: Projection on the state–space of the critical regions for stage 1 for Example 2.

DP and multi–parametric programming - instead it solves a convex multi–parametric programming for each stage of the DP procedure. Based on robust optimization, a robust reformulation of the constraints of the problem was presented which ensures that the obtained control law is a robust solution for the explicit/multi–parametric MPC.
Figure 3.5.: Projection on the state–space of the critical regions for stage 0 for Example 2.

Figure 3.6.: Simulation of the states and input for Example 2.
4. Robust
Explicit/multi-parametric Model Predictive Control for Box-constrained Linear Dynamic Systems

Robust explicit/multi-parametric controllers are designed for constrained, linear discrete–time systems with box-constrained states and inputs, involving uncertainty in the left-hand side (LHS) of the Model Predictive Control (MPC) optimization model. Based on the work of Robust explicit model predictive control algorithm, this chapter presents a reformulation that simplifies and reduce the complexity of the algorithm.

4.1. Introduction

Robust Model Predictive Control (MPC) has received significant attention in the control research literature due to its ability to account explicitly for the uncertainties in the controlled process (Zafiriou, 1990; Bemporad and Morari, 1999; Rawlings and Mayne, 2009, see). On the other hand explicit/multi–parametric MPC, which has also received significant attention in the open literature, is a control method where the nominal online MPC optimization problem is solved off–line with multi–parametric programming methods to obtain the optimal controls and objective function as explicit functions of the system states.

In this chapter, a novel method is proposed , based on the work presented in chapter 3, for the explicit/multi–parametric MPC of “boxed–constrained” linear discrete–time systems, i.e. when the state and input constraints are described by upper and lower bounds of the state and control variables.
The formulation of the Robust Explicit Model Predictive Control (MPC) problem is as follows

\[
V^*(x) = \min_{U} J(U, x)
\]

\[
= \min_{U} \sum_{t=0}^{N-1} \{x_t^T Q x_t + u_t^T R u_t\} + x_N^T P x_N
\]  \hspace{1cm} (4.1)

s.t. \hspace{0.5cm} x_{t+1} = A x_t + B u_t  \hspace{1cm} (4.2)

\[
A = A_0 + \Delta A, \quad B = B_0 + \Delta B
\]  \hspace{1cm} (4.3)

\[
\forall \Delta A \in A, \quad \Delta B \in B
\]  \hspace{1cm} (4.4)

\[
x_t \in X = \{x \in \mathbb{R}^n \mid x_{\min} \leq x \leq x_{\max}\}
\]  \hspace{1cm} (4.5)

\[
u \in U = \{u \in \mathbb{R}^m \mid u_{\min} \leq u \leq u_{\max}\}
\]  \hspace{1cm} (4.6)

\[
t = 0, 1, \ldots, N - 1
\]

\[
x_N \in X_f, \quad x = x_0
\]  \hspace{1cm} (4.7)

where \(N\) is the prediction time horizon, \(x_0\) is the initial state, \(U = [u_0^T \ldots u_{N-1}^T]\) is the sequence of current and future control variables, \(Q, P \succeq 0\) and \(R > 0\) are symmetric matrices, (4.5) and (4.6) are the state and input constraints and \(X_f = \{x \in \mathbb{R}^n \mid T x \leq \tau\}\) is the terminal constraint set. The system (4.2) is uncertain in that the system matrices \(A, B\) are given by (4.3) where \(A_0, B_0\) are of known constant values but the values of matrices \(\Delta A, \Delta B\) are not known but are bounded and given by

\[\Delta A \in A = \{\Delta A \in \mathbb{R}^{n \times n} \mid -\epsilon_a |A_0| \leq \Delta A \leq \epsilon_a |A_0|\}\]

\[\Delta B \in B = \{\Delta B \in \mathbb{R}^{n \times n} \mid -\epsilon_\beta |B_0| \leq \Delta B \leq \epsilon_\beta |B_0|\}\]

where \(\epsilon_a, \epsilon_\beta \in [0, 1]\). The objective is to obtain the control sequence \(U\), and in extension the control variable \(u_t\), as explicit functions of the state variable \(x_t\), such that the state and input constraints are satisfied for all values of the uncertain matrices \(\Delta A \in A, \Delta B \in B\). Such a solution of the explicit/multi–parametric MPC problem will be called a robust solution.
4.2. A method for Robust Explicit/Multi–parametric MPC

The proposed approach for the solution of the robust explicit MPC problem (4.1) is realized in three main steps

1. dynamic programming step: the MPC optimization (4.1) is recast in a multi–stage optimization setting
2. robust reformulation of the constraints: the optimization constraints at each stage of the resulting multi–stage problem are reformulated to account for the worst–case uncertainty, and
3. multi–parametric programming: the reformulated stage optimization problems are solved as multi–parametric problems

These steps of the proposed procedure are presented in the following sections.

4.2.1. Dynamic programming – multi–stage optimization

Following the method presented in chapter 3, problem (4.1) can be expressed as a multi–stage optimization problem and can be decomposed into a set of stage-wise problems of smaller dimensions(Bellman, 2003; Bertsekas, 2005)

\[ V_t(x_t) = \min_{u_t \in U} \sum_{i=t}^{N-1} \{ x_i^T Q x_i + u_i^T R u_i \} + x_N^T P x_N \] (4.8)

s.t. \[ x_{i+1} = A x_i + B u_i, \ i = t, \ldots, N \]
\[ \forall \Delta A \in A, \ \Delta B \in B \]
\[ x_{\min} \leq x_t \leq x_{\max}, \ u_{\min} \leq u_t \leq u_{\max} \]
\[ x_{\min} \leq x_{t+1} \leq x_{\max} \]
\[ x_{t+1} \in \mathcal{X}^{t+1} \]

where the optimization variable for (4.8) is only the current input variable \( u_t \) and only the state and input constraints at times \( t \) and \( t+1 \) are considered. The smaller problem (4.8) is solved at each stage \( t \), starting from \( t = N - 1 \) and ending at \( t = 0 \), to derive the control inputs \( u_0, \ldots, u_{N-1} \), instead of
solving the multi–stage problem as a single large–scale optimization problem. The set \( X^{t+1} \) is the set of all states \( x_{t+1} \) for which a solution \( u_{t+1} \in \mathcal{U} \) exists for the problem (4.8) at stage \( t+1 \) and is known as the feasibility set. If a control input \( u_t \) exists that satisfies the constraint \( x_{t+1} \in X^{t+1} \) then a control input \( u_{t+1} \) exists that satisfies the constraints of problem (4.8) for stage \( t+1 \). The use and method for obtaining the feasibility set will be further discussed in Section 2.5.

In Faísca et al. (2008) it was shown that (4.8) can be solved as a multi–parametric Quadratic Programming (mp–QP) problem for the case \( \Delta A = \Delta B = 0 \). However, the presence of the uncertain matrices \( \Delta A, \Delta B \) do not allow for the use of the known multi–parametric programming methods to solve (4.8) and derive the input variable \( u_t \) as an explicit function of the state. It will be shown in Sections 2.4, 2.5 a procedure to reformulate (4.8) to an mp–QP problem and derive the input variable as an explicit function of the state.

It is first considered the following state and input transformations

\[
\bar{x} = x - x_{\text{min}}, \quad \bar{u} = u - u_{\text{min}} \tag{4.9}
\]

By substituting (4.9) in the optimization problem (4.8) it is obtained the following transformed optimization problem

\[
V_t(\bar{x}_t) = \min_{\bar{u}_t \in \mathcal{U}} \sum_{i=t}^{N-1} \left\{ (\bar{x}_i + \bar{x}_{\text{min}})^T Q (\bar{x}_i + \bar{x}_{\text{min}}) \right. \\
+ (\bar{u}_i + \bar{u}_{\text{min}})^T R (\bar{u}_i + \bar{u}_{\text{min}}) \\
+ (\bar{x}_N + \bar{x}_{\text{min}})^T P (\bar{x}_N + \bar{x}_{\text{min}}) \right\} \\
\text{s.t.} \quad \bar{x}_{i+1} = A\bar{x}_i + B\bar{u}_i + \bar{g}, \quad i = t, \ldots, N \tag{4.10}
\]

\[
0 \leq \bar{x}_t \leq x_{\text{max}} - x_{\text{min}} \tag{4.12}
\]

\[
0 \leq \bar{x}_{t+1} \leq x_{\text{max}} - x_{\text{min}} \tag{4.13}
\]

\[
0 \leq \bar{u}_t \leq u_{\text{max}} - u_{\text{min}} \tag{4.14}
\]

\[
\bar{x}_{t+1} \in \mathcal{X}^{t+1} \tag{4.15}
\]

for all \( \Delta \in \mathcal{A}, \Delta B \in \mathcal{B} \) and where \( \bar{g} = (A-I)x_{\text{min}} + Bu_{\text{min}} \) and the set \( \mathcal{X}^{t+1} \) is obtained by substituting (4.9) into the feasibility set \( X^{t+1} \). Note that the objective function in (4.10) is a \textit{convex quadratic function} of \( \bar{x} + x_{\text{min}} \) and
\( \bar{u} + u_{\min} \) and hence the minimization in (4.10) forces \( \bar{x}_k \to -x_{\min} \) and \( \bar{u}_k \to -u_{\min} \) and therefore, from relations (4.9), \( x_k \to 0 \) and \( u_k \to 0 \).

**Remark 25.** Note that if the solution of (4.10) is \( \bar{u}_t \) then the solution of (4.8) is \( u_t = \bar{u}_t + u_{\min} \).

Furthermore, an mp–QP formulation of (4.10) can be derived by (i) considering \( u_t \) as the optimization variable, (ii) considering \( \bar{\theta}_t = [x^T_t \ u^T_{t+1} \ldots \ u^T_{N-1}] \) as the vector of parameters, (iii) combining the feasibility constraint \( \bar{X} \) and the state constraints \( 0 \leq \bar{x}_t \leq x_{\max} - x_{\min} \) into the inequality constraints

\[
G^{t+1}x_{t+1} \leq b^{t+1}
\]

(iv) incorporating the linear system model (4.11) into the constraints, and (v) incorporating the nominal system dynamics ((4.11) with \( \Delta A = \Delta B = 0 \)) in the objective function (objective penalizes the nominal system performance). Following the above steps it is obtained the following multi–parametric programming problem

\[
V_t(\bar{x}_t) = \min_{\bar{u}_t \in \mathcal{U}} \left\{ \frac{1}{2} \bar{u}^T_t H \bar{u}_t + \bar{\theta}^T_t F \bar{u}_t + L^T_u \bar{u} \right\}
\]

\[
+ \bar{\theta}^T_t Y \bar{\theta}_t + L^T_\theta \bar{\theta} + c
\]

\[
\text{s.t. } G^{t+1}A_0 \bar{x}_t + G^{t+1} \Delta A \bar{x}_t + G^{t+1}B_0 \bar{u}_t + G^{t+1} \Delta B \bar{u}_t + G^{t+1} \bar{g} \leq b^t
\]

\[
0 \leq \bar{x}_t \leq x_{\max} - x_{\min}
\]

\[
0 \leq \bar{u}_t \leq u_{\max} - u_{\min}
\]

where \( \Delta A \in \mathcal{A} \) and \( \Delta B \in \mathcal{B} \). The matrices \( H, F, L_u, Y, L_\theta \) and \( c \) are of appropriate dimensions and are obtained after substituting the nominal system model in the objective function of (4.10). Note that since the objective function of (4.10) is a convex quadratic function of \( \bar{u}_i \) and \( \bar{x}_i \), \( i = t, \ldots, N - 1 \), the objective function of (4.17) is also a convex quadratic function of \( u_i \), \( i = t, \ldots, N - 1 \) and \( x_0 \). Note also that the matrix coefficients in (4.18) are uncertain, hence problem (4.17) is a robust mp–QP problem. In order for a control input \( \bar{u}_t \) to be a robust solution of (4.17), the constraint (4.18) has to be satisfied for all values of the uncertainty.
4.2.2. Robust Reformulation

In order to ensure that the constraints of (4.17) are satisfied at all stages \( t \) and for all possible values of the uncertain matrices \( \Delta A, \Delta B \), it is applied the following robust reformulation (Ben-Tal and Nemirovski, 2000). From equation (4.18) we have

\[
G^{t+1}A_0\bar{x}_t + G^{t+1}B_0\bar{u}_t + G^{t+1}\bar{g} \leq b^{t+1}
\]  
(4.19)

\[
G^{t+1}A_0\bar{x}_t + \epsilon_a |G^{t+1}|A_0||\bar{x}_t| + G^{t+1}B_0\bar{u}_t + \epsilon_b |G^{t+1}|B_0||u_t| + G^{t+1}\bar{g} \leq b^{t+1} + \delta \max\{1, |b^{t+1}|\}
\]  
(4.20)

where inequality (4.19) ensures that the problem is feasible for the nominal system while inequality (4.20) represents the realization of the first constraint in (4.17) for the worst-case value of the uncertainty. The variable \( \delta \) is a measure of the tolerated infeasibility i.e. how much the constraint can be relaxed to ensure a feasible solution. Obviously, no infeasibility is allowed when \( \delta = 0 \).

The inequality (4.20) is nonlinear with respect to \( \bar{x} \) and \( \bar{u} \) and hence replacing it in (4.17) will result in a multi-parametric nonlinear programming problem. However, since from (4.12), (4.14) it known that \( \bar{x}_t \geq 0, \bar{u}_t \geq 0 \), can be replaced the absolute values \( |\bar{x}_t| \) and \( |\bar{u}_t| \) in (4.20) by \( \bar{x}_t, \bar{u}_t \) and re-write the inequality (4.20) as a linear inequality of \( \bar{x}, \bar{u} \)

\[
G^{t+1}A_0\bar{x}_t + \epsilon_a |G^{t+1}|A_0|\bar{x}_t| + G^{t+1}B_0\bar{u}_t + \epsilon_b |G^{t+1}|B_0|u_t| + G^{t+1}\bar{g} \leq b^{t+1} + \delta \max\{1, |b^{t+1}|\}
\]  
(4.21)

(4.21) is then substituted in (4.17) to obtain the following mp-QP formu-
where \( \bar{u}_t \) is the optimization variable and \( \bar{\theta}_t \) is the vector of parameters.

Problem (4.17) is a mp–QP reformulation of the stage optimization problem (3.20).

If \( \bar{u}_t \) is a solution for (4.22) then \( \bar{u}_t \) satisfies (4.19)–(4.20) and hence it satisfies the constraint (4.18) for all values of \( \Delta A, \Delta B \). This implies that \( \bar{u}_t \) is a robust solution for (4.17) and hence for (4.10). In addition, since (4.10) is obtained by applying the linear transformation (4.9) on (3.20), then \( u_t = \bar{u}_t + u_{\min} \) is also a robust solution of (3.20). It can now stated the following Lemma

**Lemma 4.** If \( \bar{u}_t \) is a feasible solution for (4.22) then it is also a robust solution for (4.10) and \( u_t = \bar{u}_t + u_{\min} \) is a robust solution for (3.20).

**Remark 26.** The nonlinear inequality (4.20) can be relaxed to the set of linear inequalities \( G^{t+1} A_0 \bar{x}_t + G^{t+1} B_0 \bar{u}_t + G^{t+1} \bar{g} \leq b^{t+1} \)

\[
G^{t+1} A_0 \bar{x}_t + c_a |G^{t+1}| A_0 |\bar{x}_t + G^{t+1} B_0 \bar{u}_t + c_b |G^{t+1}| B_0 |\omega_t + G^{t+1} \bar{g} \leq b^{t+1} + \delta \max\{1, |b^{t+1}|\},
\]

\[
0 \leq \bar{x}_t \leq x_{\max} - x_{\min}, 0 \leq \bar{u}_t \leq u_{\max} - u_{\min}
\]

by introducing two new optimization variables \( z_t, \omega_t \) and four extra inequalities, thus increasing the number of constraints in (4.22). However, as has been shown above, this is not anymore necessary since \( \bar{x}, \bar{u} \) are positive and (4.20) can be replaced only by (4.21) without increasing the number of constraints in (4.22).

### 4.2.3 Multi–parametric procedure

Since (4.22) is an mp–QP problem, the solution to (4.22) is given by the following explicit form (Pistikopoulos and Dua, 2007b,a)

\[
\bar{u}_t = K_t \bar{\theta}_t + c_t; \text{ if } \bar{\theta}_t \in CR_i, CR_i = \{ \bar{\theta}_t \mid H_i^T \bar{\theta}_t \leq h_i \}
\]
\[ \bar{u}_t = f_t^*(\bar{\theta}_t) = f_t^*(\bar{x}_t, \bar{u}_{t+1}, \ldots, \bar{u}_{N-1}) \]  \hspace{1cm} (4.24) \\

where \( i = 1, \ldots, s_t \), \( K_t^i, c_t^i \) are matrices of appropriate dimensions and \( CR_t^i \subset \mathbb{R}^n \) are the corresponding critical regions. The expression (4.23) describes the relation between the solution \( \bar{u}_t \) at the current stage and the solutions \( \bar{u}_{t+1}, \ldots, \bar{u}_{N-1} \) at the previous stages. However, our objective is to obtain the input \( \bar{u}_t \) as an explicit function of the incumbent state \( \bar{x}_t \). Hence, in the following it is presented a procedure for deriving i) \( \bar{u}_t \) as an explicit function of the state \( \bar{x}_t \) and ii) the feasibility set \( \bar{X}^t \).

Reduction of the mp–QP solution:

It will be demonstrated the procedure for deriving an expression \( u_t = f_t^*(x_t) \) from (4.23) for the stages \( t = N - 1 \) and \( t = N - 2 \). For the stages \( N - 1 \) and \( N - 2 \) the control variables are \( \bar{u}_{N-1} \) and \( \bar{u}_{N-2} \) while the parameters are \( \bar{\theta}_{N-1} = \bar{x}_{N-1} \) and \( \bar{\theta}_{N-2} = [\bar{x}_{N-1} \ \bar{u}_{N-1}]^T \). The expression (4.23) for \( \bar{u}_{N-1} \) and \( \bar{u}_{N-2} \) are then given by

\[
\bar{u}_{N-1} = K_{N-1}^i \bar{x}_{N-1} + c_{N-1}, \quad \text{if} \quad \bar{x}_{N-1} \in CR_{N-1}^i \hspace{1cm} (4.25) \\
\bar{u}_{N-2} = K_{N-2}^j \bar{x}_{N-2} + L_{N-2}^j \bar{u}_{N-1} + c_{N-2}, \quad \text{if} \quad \bar{x}_{N-2}, \bar{u}_{N-1} \in CR_{N-2}^j \hspace{1cm} (4.26)
\]

where \( i = 1, \ldots, s_{N-1} \) and \( j = 1, \ldots, q_{N-2} \). Note that \( \bar{u}_{N-1} \) is an explicit PWA function of the state \( x_{N-1} \) while \( \bar{u}_{N-2} \) is a function of \( \bar{x}_{N-2} \) and \( \bar{u}_{N-1} \). In order to obtain \( \bar{u}_{N-2} \) only as an explicit function of \( \bar{x}_{N-2} \), it is applied the following steps to eliminate \( u_{N-1} \) from (4.26) (Faisca et al., 2008): i) first, the system model \( \bar{x}_{N-1} = A_0 \bar{x}_{N-2} + B_0 \bar{u}_{N-2} + \bar{g} \) is incorporated in (4.25) in order to express \( \bar{u}_{N-1} \) as a function of \( \bar{x}_{N-2} \) and \( \bar{u}_{N-2} \) and ii) (4.25) and (4.26) are combined for all \( i, j \) to obtain a set of piecewise affine (PWA) expressions with respect to \( \bar{x}_{N-2}, \bar{u}_{N-2} \) and \( \bar{u}_{N-1} \)

\[
\bar{u}_{N-1} = K_{N-1}^i A_0 \bar{x}_{N-2} + K_{N-1}^i B_0 \bar{u}_{N-2} + K_{N-1}^i \bar{g} + c_{N-1} \hspace{1cm} (4.27) \\
\bar{u}_{N-2} = K_{N-2}^j \bar{x}_{N-2} + L_{N-2}^j \bar{u}_{N-1} + c_{N-2} \hspace{1cm} (4.28) \\
\bar{x}_{N-2}, \bar{u}_{N-2} \in CR_{N-1}^i, \quad \bar{x}_{N-2}, \bar{u}_{N-1} \in CR_{N-2}^j \hspace{1cm} (4.29)
\]
Then, by: i) directly substituting (4.27) in (4.28) and (4.29) and solving for \( \bar{u}_{N-2} \) or ii) using elimination methods such as orthogonal projection or Fourier-Motzkin elimination to eliminate \( \bar{u}_{N-1} \) from (4.27), (4.28) and (4.29), the control input \( \bar{u}_{N-2} \) can be obtained as an explicit function of \( \bar{x}_{N-2} \)

\[
\bar{u}_{N-2} = f^*_N(\bar{x}_{N-2})
\]

\[
\bar{u}_{N-2} = K^j_{N-2}\bar{x}_{N-2} + c^j_{N-2}, \text{ if } \bar{x}_{N-2} \in CR^j_{N-2} \tag{4.30}
\]

where \( j = 1, \ldots, s_{N-2} \) and \( CR^j_{N-2} \) is the critical region in which (4.30) is valid. Note that expressions (4.27)–(4.29) are obtained for all possible combinations \( i, j \) of the critical regions of (4.25) and (4.26), and correspond to feasible values of \( \bar{x}_{N-2}, \bar{u}_{N-2} \) and \( \bar{u}_{N-1} \) for problem (4.22). It is possible that some combination of \( i, j \) is not realizable, which implies that no feasible solutions exists.

The same procedure is applied for all stages \( t \). Let \( \bar{u}_{t+1} = f^*_{t+1}(\bar{x}_{t+1}), \ldots, \bar{u}_{N-1} = f^*_{N-1}(\bar{x}_{N-1}) \) be the solutions of the (4.22) for stages \( t + 1, \ldots, N - 1 \) and \( \bar{u}_t = f^*_t(\bar{\theta}_t) \) the solution for the stage \( t \) given by (4.24). Then, i) by replacing \( x_k \) with \( \bar{x}_k = A^k_{t-1}\bar{x}_t + \sum_{i=0}^{k-1-t} A^k_{i}B_0\bar{u}_{k-1-j} \) in the control variables \( \bar{u}_k = f^*_k(\bar{x}_k) \) for all \( k = t + 1, \ldots, N - 1 \) and ii) by combining the critical regions and control expressions of all control variables it is obtained the following set of PWA expressions on \( x_t, u_t, \ldots, u_{N-1} \)

\[
\bar{u}_{N-1} = f^*_N(\bar{x}_t, \bar{u}_t, \ldots, \bar{u}_{N-2}), \ldots, \tag{4.31}
\]

\[
\bar{u}_{t+2} = f^*_{t+2}(\bar{x}_t, \bar{u}_t, \bar{u}_{t+1}), \bar{u}_{t+1} = f^*_{t+1}(\bar{x}_t, \bar{u}_t) \tag{4.32}
\]

\[
\bar{u}_t = f^*_t(\bar{x}_t, \bar{u}_{t+1}, \ldots, \bar{u}_{N-1}) \tag{4.33}
\]

Then the variables \( \bar{u}_{t+1}, \ldots, \bar{u}_{N-1} \) are eliminated either by i) substituting (4.31)-(4.32) in (4.33) and solving for \( \bar{u}_t \) or ii) applying elimination techniques on (4.31)–(4.33), to obtain \( \bar{u}_t \) as an explicit function of the state \( \bar{x}_t \), \( \bar{u}_t = f^*_t(\bar{x}_t) \) where

\[
\bar{u}_t = K^i_t\bar{x}_t + c^i_t, \text{ if } \bar{x}_t \in CR^i_t, \text{ i } = 1, \ldots, s_t \tag{4.34}
\]

and \( CR^i_t \) is the critical region where the control (4.34) is valid.
Calculation of the feasibility constraint set:

Once the explicit solution (4.34) has been obtained, the feasibility constraint set $\bar{X}^t$ for stage $t$ can then be obtained from the following expression

$$\bar{X}^t = \bigcup_{i=1}^{s_t} \mathcal{CR}^i_t$$

(4.35)

Since (4.22) is a convex multi–parametric quadratic programming problem, the set of all critical regions in the combined PWA expressions (4.31)–(4.33) is a convex polyhedral set. Moreover, the set of all critical regions $\mathcal{CR}^i_t$ is also a convex polyhedral set, since it is obtained by performing linear operations on (4.31)–(4.33), and hence can be described by a set of linear inequalities

$$\bar{X}^t = \{ \bar{x} \in \mathbb{R}^n \mid H^t \bar{x} \leq h^t \}$$

(4.36)

**Remark 27.** The use of the feasibility constraints $\bar{x}_{t+1} \in \bar{X}^{t+1}$ is very important since it ensures that the future state $x_{t+1}$ lies in the set of states for stage $t + 1$, for which a feasible and robust control $\bar{u}_{t+1} = f^*_t(\bar{x}_{t+1})$ exists. Therefore, by applying $\bar{u}_t$ to the system (4.11), it will guarantee that a feasible control $\bar{u}_{t+1} = f^*_t(\bar{x}_{t+1})$ can be obtained at the next time $t + 1$.

The control input $u_t$ can then be obtained by substituting (4.9) in (4.34)

$$u_t = \mu_t(x_t) = \bar{u}_t + f^*_t(x_t - x_{min})$$

(4.37)

Note from (4.37) that $u_t$ is also an explicit function of the state $x_t$. Specifically, $\mu_t(x_t)$ is a PWA function of $x_t$ since $f^*_t(x_t - x_{min})$ is a PWA function of $x_t$. Finally, the feasibility set $\bar{X}^{t+1}$ can be obtained by substituting $\bar{x} = x - x_{min}$ in (4.36).

**Remark 28.** Note that the proposed algorithm, although it is based on DP methods, does not follow the conventional DP approaches (Bemporad et al., 2003). If conventional DP methods were used, then the solutions $\bar{u}_{t+1} = f^*_{t+1}(\bar{x}_{t+1}), \ldots, \bar{u}_{N-1} = f^*_{N-1}(\bar{x}_{N-1})$ from the previous stages should be first incorporated in the formulation of problem (4.22) which would then become a non-linear multi–parametric programming problem, thus requiring a global optimization procedure for its solution (Faísca et al., 2008). On the other hand, the proposed procedure takes into account the convexity of (4.22) with
respect to the control variables $\bar{u}_i$, $i = t, \ldots, N - 1$ and state $x_t$, to solve an mp–QP problem (4.22) at each stage. The explicit solution (4.34) is then derived by performing a set of linear algebraic manipulations ((4.31)–(4.33)).

4.2.4. Algorithm for robust explicit/multi–parametric MPC for linear system with “box”–constraints

Based on the proposed procedure that was described in Sections 4.2.3 it can now be proposed a DP–based algorithm for robust explicit/multi–parametric programming which is shown in Table 4.1. In Step 1 of the proposed algorithm, problem (4.22) is solved for stage $t = N - 1$ and $u_{N-1} = f_{N-1}(x_{N-1})$ is obtained as a function of $x_{N-1}$. The algorithm then proceeds iteratively by applying Steps 2i. – 2iv. for each stage $t$. In Step 2i the mp–QP problem (4.22) is solved to obtain $\bar{u}_t = f^*_t(\theta_t)$. In Step 2ii the reduction procedure described in section 2.5 is applied and the control variable $\bar{u}_t$ is obtained as an explicit function of the state $\bar{u}_t = f^*_t(x_t)$. In step 2iii. the control input $u_t = \mu_t(x_t)$ is obtained from expression (4.37) and finally in step 2iv. the feasibility set $\bar{X}_t$ is obtained from (4.35). The algorithm then proceeds to the next stage $t - 1$ and terminates when stage $t = 0$. After repeating all the steps of the proposed algorithm it is obtained a sequence of control laws $U = \{u_t, \ldots, u_{N-1}\} = \{\mu_0(x_0), \ldots, \mu_{N-1}(x_{N-1})\}$.

Each of the control inputs $u_t = \mu_t(x_t)$, $t = 0, \ldots, N - 1$ is a robust solution of (3.20), hence the constraints of (3.20) are satisfied for all values of the uncertain matrices $\Delta A, \Delta B$ (see Lemma 4). This also implies that the control inputs $u_t = \mu_t(x_t)$, $t = 0, \ldots, N - 1$ satisfy the state and inputs constraints (4.5)–(4.6) for all values of $\Delta A, \Delta B$ which are also the constraints for (4.1). Therefore, the control sequence $U$ satisfies the constraints of the explicit/multi–parametric MPC problem (4.1) for all values of $\Delta A, \Delta B$.

Lemma 5. The control sequence $U = \{u_t, \ldots, u_{N-1}\}$, where $u_t = \mu_t(x_t)$, $t = 0, \ldots, N - 1$, is a robust solution of the explicit/multi–parametric MPC problem (4.1).
Key features of the proposed robust explicit/multi–parametric MPC method:

The key features of the proposed robust explicit MPC method are summarized as follows: (i) the proposed method obtains a sequence of robust explicit control laws $u_t = \mu_t(x_t)$, $t = 0, \ldots, N - 1$ which guarantee that the state and input constraints are always satisfied for all values of the uncertainty (Lemma 5), (ii) a convex mp-QP problem is solved for each stage of the proposed DP procedure, without the need to solve a Global Optimization problem (Remark 28), and (iii) the number of optimization variables and constraints for each stage optimization problem (4.22) is not increased compared to the previous methods (Faisca et al., 2008) (Remark 26).

Note that the proposed algorithm was developed for the case of linear discrete–time system with the state and input constraints described by (4.5)–(4.6). Note also that at each stage $t$ of the proposed algorithm, the number of parameters $\bar{\theta}_t = \{\bar{x}_t, \bar{u}_{t+1}, \ldots, \bar{u}_{N-1}\}$ involved in (4.22) increases as $t$ decreases from $N - 1$ to 0. Also the number of expressions (4.31)–(4.33) increases as $t$ decreases hence increasing the complexity of the algebraic manipulations required to eliminate $\bar{u}_{t+1}, \ldots, \bar{u}_{N-1}$ from (4.31)–(4.33) and derive the control law (4.34). In addition if a relatively larger number of critical regions is obtained at each stage for (4.34) then this might increase the complexity of the calculations for obtaining the feasibility set from (4.35).

On–line Implementation:

The on–line implementation of the explicit/ multi–parametric MPC can be realized with two approaches. In the first approach, the whole control sequence $U = \{u_0, u_1, \ldots, u_{N-1}\}$ is applied to the system, with each control input $u_t = \mu_t(x_t)$ applied at the corresponding time instant $t$. With the second approach, only the first control $u_0 = \mu_0(x_0)$ of the control sequence $U$ is applied at each time $t$, by considering the current state $x$ as the initial state $x_0$.

4.3. Concluding Remarks

A new algorithm for robust explicit/multi–parametric MPC was presented for the case of linear systems with state and input constraints described
by upper and lower bounds on the state and input variables. The algo-
rithm features three key steps, based on DP, robust optimization and multi–
parametric programming methods and allows for the derivation of robust explicit control solutions to the robust explicit MPC problem. The main advantage of this algorithm is that switches the range of the input and state variables in order to replace the absolute values of the original optimisation problem with a simpler reformulation. This leads to a less complicated solution and consequently to smaller computation time. I have solved the same examples as in chapter 3 and provided the same explicit solution, thus I didn’t re-write them in this chapter.
Table 4.1: Algorithm for Robust Explicit Multi–Parametric MPC

**Step 1.** Set $k = N - 1$: solve the mp-QP problem (4.22) with $x_{N-1}$ being the parameters and obtain $u_{N-1} = f^*_x(x_{N-1})$ (Eq. (4.25)) and $X^{N-1}$ (Eq. (4.36)).

**Step 2.** Set $k$ to the current stage:

1. solve the $k^{th}$ stage-wise mp–QP problem (4.22) with $\bar{x}_k, \bar{u}_k, \ldots, \bar{u}_{N-1}$ being the parameters and obtain $\bar{u}_k = f^*_k(\bar{x}_k, \bar{u}_k, \ldots, \bar{u}_{N-1})$,
2. obtain $\bar{u}_k = f^*_k(\bar{x}_k)$ by eliminating $\bar{u}_{k+1}, \ldots, \bar{u}_{N-1}$ from (4.31)–(4.33),
3. obtain $u_t = \mu_t(x_t)$ from (4.37)
4. calculate the feasibility set $X^k$ from Eq. (4.35)

**Step 3.** Set $k = k - 1$: if $k = 0$ stop, else go to Step 2.
Part II.

Energy Systems Application
5. Dynamic Optimization and robust explicit model predictive control of hydrogen storage tank

In this chapter we present a detailed two-dimension dynamic mathematical model and an explicit/multi-parametric controller for metal hydride bed reactor. For the controller design, a reduced order approximate model is obtained, based on which nominal and robust multi-parametric controllers are designed.

5.1. Introduction

The depletion of fossil fuel natural resources and the increase of environmental hazards resulting from the use of such fuels have led to significant research effort in alternative and cleaner fuels (Aldas and Kaplan, 2002). Hydrogen appears to be a good candidate as a fuel carrier for industrial and mobile applications since it can be produced from renewable energy sources. Hydrogen has high calorific value, the highest energy content per unit weight than any other fuel and is environmentally friendly (Dhaou and Nasrallah, 2006). However, the transition from the use of fossil fuels to hydrogen poses a number of challenges, especially in the development of effective hydrogen storage mechanisms.

Technologies that have been developed for the storage of hydrogen for mobile devices, such as compression and liquefaction (Momirlan and Veziroglu, 2002) are in general impractical due to high energy requirements and space limitations. An alternative technology, using metal-hydride alloys (LaNi5) capable to adsorb and release large amounts of hydrogen by bonding with forming hydrides, has been developed (Jemni and Nasrallah, 1995) over the
years to partially overcome these issues. In this respect, modelling, optimization and control of such systems can play a crucial role in assessing their economic and operability performance, as reliable hydrogen storage alternatives.

Several studies have appeared in the open literature in the area of modelling and optimization of such systems, including one-dimensional (Choi and Mills (1990), Gopal and Murty (1992)), two-dimensional (Mayer et al. (1987), Bhouri et al. (2000), Askri (2003), Demircan et al. (2005), Muhittin and Ercan (2005), Kikkinides et al. (2006b)) and three-dimensional (Aldas and Kaplan (2002) and Mat et al. (2002)) models.

Only a few recent studies have appeared in the open literature addressing the control of the adsorption/desorption process of the metal hydride bed process. Kikkinides et al. (2006), Kikkinides et al. (2006b) and Georgiadis et al. (2008) presented systematic approaches for the optimization and open loop control of tubular metal-hydride beds with additional cooling heat exchangers. It was shown that by appropriately controlling the thermal dynamics of the reactor, the adsorption rate of hydrogen in the process can be significantly improved.

In this chapter, it is presented a systematic framework for the optimal design, optimization and advanced control of the desorption process in metal-hydride bed reactors. A detailed mathematical model is first presented, based on the earlier works of Georgiadis et al. (2009). Design optimization and dynamic simulation and optimization studies are then performed, based on which a reduced order state-space (SS) model, suitable for the design of advanced model-based controllers, is derived. Both nominal and robust mp-MPC controllers are designed, validated and compared to an optimized PI controller.

The methodology, that was used to design the controllers is described below,

1. Development of a high fidelity mathematical modeling of the metal hydride reactor desorption process - used for detailed simulation and (design and operational) optimization studies.

2. Development of a reduced order/approximating model, suitable for multi-parametric MPC

3. Design of nominal & robust multi-parametric MPC controllers
4. Validation of the controllers

Step 1 involves the development of a high-fidelity mathematical model, for performing detailed dynamic simulation and design/operational optimization studies. The model is validated using experimental data in several operation conditions in order to guarantee the accuracy of the simulation results. In this work, it is employed the 2D mathematical model developed by Kikkinides et al. (2006b). In step 2, a reduced order approximated model is derived by performing system identification on the simulation data. Step 3 corresponds to the design of the multi-parametric/explicit Model Predictive Controllers (mp-MPC), by applying the available theory and tools of multi-parametric programming and control (Pistikopoulos and Dua (2007b), Pistikopoulos and Dua (2007a)). Finally step 4 involves the offline validation of the derived multi-parametric/explicit controllers. These are described next.

5.2. Mathematical Model of the Metal-Hydride Bed

This study involves a cylindrical metal-hydride reactor as shown in Figure 5.1. The reactor is filled with metal hydride alloy, namely LaNi$_5$, surrounded by a jacket heat exchanger. Additionally, two heat exchangers are included in the reactor to improve adsorption/desorption process. A concentric tube at the center of the reactor from $r=0$ to $r=r_0$ and a concentric annual ring of thickness $\delta r = r_2-r_1$ filled with a flowing heating fluid. The geometric position of the concentric annual ring is variable and directly affects the performance of the metal hydride. However, Kikkinides et al. (2006b) proved that for a metal hydride bed without an expansion volume the optimal position is at $r=0.5$. The 2D mathematical model for the desorption process and the integrated heat exchangers is based on the original contribution of Kikkinides et al. (2006b) under the following assumptions:

1. at the beginning of the process the hydride bed is filled with hydrogen and the bed temperature coincides with the inlet temperature
2. ideal gas law holds in the gas phase
3. local thermal equilibrium is valid
radiative heat transfer is negligible

5. axial and radial dispersion are included in the mass balances for interstitial fluid

6. the axial and radial pressure drops in the reactor depend linearly on the fluid velocity through Darcy’s law

7. thermophysical properties are constant

Here, I extend further this model without the assumption of local thermal equilibrium between gas and solid phase (assumption 3) – this allows us to compute the actual temperature (profile) of the porous medium and hydrogen in the metal hydride. The partial differential and algebraic equations (PDAE) governing the desorption process are the mass balance of porous media and hydrogen, momentum balance of hydrogen and energy balance equations of porous media, hydrogen and heat exchangers.

5.2.1. Mass balance of hydrogen in the gas phase (bulk of the reactor)

\[
\frac{\partial \hat{\rho}}{\partial \tau} + \frac{1}{\varepsilon} \nabla \cdot (\hat{\rho} \hat{u}) - \left( \frac{1}{P_{\text{em}}} \right) \cdot \nabla^2 \hat{\rho} + W_0 \cdot \frac{\partial \hat{\rho}_s}{\partial \tau} = 0
\]  

(5.1)

where \( \tau \) is the dimensionless time, \( \tau = u_0 / L \) (\( L \) is the bed length and \( u_0 \) is a reference velocity), \( \hat{u} = (\hat{u}_z, \hat{u}_r) \) is the superficial gas velocity vector,
\( \hat{\rho} = \rho / \rho_{e} \) is the density of hydrogen in the gas phase, \( \rho_{e} \) being the density at the equilibrium, and \( \hat{\rho}_{s} \) is the solid (metal) density of the alloy. \( Pe_{m} \) and \( W_{0} \) are dimensionless parameters defined below and \( \varepsilon \) is the void fraction of the bed.

### 5.2.2. Pressure drop equations

The steady state momentum balance of laminar gas flow through a packet bed can be expressed by Darcy’s law

\[
\hat{u}_{z} = -K_{z} \frac{\partial \hat{P}}{\partial z}, \quad \hat{u}_{r} = -K_{r} \frac{\partial \hat{P}}{\partial r}
\]

where \( K_{z}, K_{r} \) are the dimensionless permeability values of the bed and \( P_{0} \) is the equilibrium pressure, variables \( z = z^* / L, \ r = r^* / R \) are the dimensionless axial and radial distance, normalized by the total length \( L \) and bed radius \( R \). It is assumed that in Eq. (5.2) the transient and inertial terms are negligible (Yang et al., 1993).

### 5.2.3. Energy balance equations

Since assumption 3 is not any more considered in this model and since there is not thermal equilibrium in the reactor, the energy balances are described by the following two equations.

For the fluid:

\[
\hat{\rho} \cdot \frac{\partial \theta}{\partial \tau} + \left( \frac{\varepsilon}{\hat{\rho}} \right) \cdot \hat{u} \cdot \nabla \theta - \left( \frac{1}{P_e} \right) \cdot \nabla^2 \theta - P_{e_{g}} (\theta - \theta_{s}) = 0
\]  

(5.3)

For the solid:

\[
\hat{\rho}_{s} \cdot \frac{\partial \theta_{s}}{\partial \tau} - E_{s} \cdot \nabla^2 \theta_{s} - P_{e_{s}} (\theta - \theta_{s}) - W_{0} \cdot \frac{\partial \hat{\rho}_{s}}{\partial \tau} \left[ \beta + \left( 1 - \frac{C_{ps}}{C_{pg}} \right) \theta \right] = 0
\]

(5.4)

where \( \theta = T / T_{0} \) is the gas temperature, \( \theta_{s} = T_{s} / T_{0} \) is the solid gas temperature, \( T_{c} \) being the temperature of the equilibrium state and \( \theta_{c} \) is the inlet temperature of the heating medium that is constant in time and space.

The heat exchanger coefficient between solid and gas is given as Jemni et al.
where \( \bar{d}_p \) is the average particle diameter and, \( Re \) and \( Pr \) are the Reynolds and Prandtl number, respectively.

5.2.4. Mass balance of the metal hydride

\[
(1 - \varepsilon) \cdot \frac{\partial \hat{\rho}_s}{\partial \tau} = C_{a} \cdot t_{res} \cdot \exp \left( -\varepsilon_d / \theta \right) \cdot \ln \left( \frac{\hat{P}}{\hat{P}_{eq}} \right) \cdot (\hat{\rho}_{sat} - \hat{\rho}_s) \tag{5.6}
\]

where \( \hat{P}_{eq} \) is the dimensionless equilibrium pressure, \( \hat{\rho}_{sat} \) is the saturated bed density and \( \varepsilon_d \) is the dimensionless activation energy of the reaction kinetics process.

5.2.5. Definition of equilibrium pressure

\[
P_{eq} = f \left( \frac{H}{M} \right) \cdot \exp \left[ \frac{\Delta H}{R_g} \cdot \left( \frac{1}{T_0 \theta} - \frac{1}{T_{ref}} \right) \right] \tag{5.7}
\]

Where the function \( f \left( \frac{H}{M} \right) \) is the equilibrium pressure at the reference temperature \( T_{ref} \) and is usually obtained by fitting experimental data to a polynomial of nth degree (Kikkinides et al. (2006b)).

The equations and variables of the model, presented so far, are in dimensionless form. Hence several dimensionless parameters are introduced which are given below

\[
W_0 = \left( \frac{1 - \varepsilon}{\varepsilon} \right) \frac{\rho_s 0}{\rho_0} , \quad Le = \frac{W_0 \cdot C_{ps}}{C_{pg}} , \quad \beta = \frac{-\Delta H}{C_{pg} \cdot MW \cdot T_0}
\]

\[
K_z = \frac{K_0 \cdot P_0}{\mu \cdot u_0 \cdot L} , \quad K_r = \frac{K_0 \cdot P_0}{\mu \cdot u_0 \cdot R} , \quad \varepsilon_d = \frac{E_d}{R_g T_0} , \quad t_{res} = \frac{L}{u_0} \lambda_e = \varepsilon \lambda_g + (1 - \varepsilon) \lambda_g ,
\]

\[
P_{et,z} = \frac{\varepsilon \cdot C_{H_2O} \cdot u_0 \cdot C_{pg} \cdot L}{\lambda_e} , \quad P_{et,r} = \frac{\varepsilon \cdot C_{H_2O} \cdot u_0 \cdot C_{pg} \cdot R}{\lambda_e} , \quad P_{em,z} = \frac{\varepsilon u_0 L}{D_z} , \quad P_{em,r} = \frac{\varepsilon u_0 R}{D_r} , \quad \text{Bi}_z = \left( \frac{hL}{\lambda_e} \right) , \quad \text{Bi}_r = \left( \frac{hR}{\lambda_e} \right)
\]

\[
P_{es} = \frac{L_0 \cdot H_g}{u_0 \cdot (1 - \varepsilon) \cdot C_{ps} \cdot p_0} , \quad P_{eg} = \frac{L_0 \cdot H_g}{u_0 \cdot (1 - \varepsilon) \cdot C_{pg} \cdot p_0}
\]
5.2.6. Modelling the heat exchange process

An energy balance for each heat exchanger inside the reactor has also to be introduced

\[ \frac{\partial \theta f_1}{\partial \tau} - \frac{1}{P e_{h1}} \cdot \frac{\partial^2 \theta f_1}{\partial z^2} + \hat{u} f_1 \frac{\partial \theta f_1}{\partial z} + B_i M \cdot (\theta f_1 - \theta(z, r_0)) = 0 \]

\[ \frac{\partial \theta f_2}{\partial \tau} - \left( \frac{1}{P e_{h2}} \right) \cdot \frac{\partial^2 \theta f_2}{\partial z^2} + \hat{u} f_2 \frac{\partial \theta f_2}{\partial z} + B_i M \cdot (\theta f_2 - \theta(z, r_1)) + B_i M \cdot (\theta f_2 - \theta(z, r_2)) = 0 \]

(5.8)

5.2.7. Boundary and initials conditions

The necessary boundary and initials conditions to complete the IPDAEs of the model are given next.

At the tank inlet (Dirichlet boundary conditions, \( z=0 \)):

\[ \hat{\rho} = 1, \ \theta = 1, \ \hat{P} = 1 \] (5.9)

At the tank outlet (\( z=1 \)):

\[ \frac{\partial \hat{\rho}}{\partial z} = 0, \ -\frac{\partial \theta}{\partial z} = B_i z \cdot (\theta - \theta_c), \ \hat{u}_z = \hat{u}_r = 0 \] (5.10)

At the inner tube (\( r=r_0 \)):

\[ \frac{\partial \hat{\rho}}{\partial r} = 0, \ \frac{\partial \theta}{\partial r} = B_i r \cdot (\theta - \theta_c), \ \hat{u}_z = \hat{u}_r = 0 \] (5.11)

At the concentric ring (\( r=r_1 \)):

\[ \frac{\partial \hat{\rho}}{\partial r} = 0, \ -\frac{\partial \theta}{\partial z} = B_i \cdot (\theta - \theta_c), \ \hat{u}_z = \hat{u}_r = 0 \] (5.12)

At the concentric annular ring (\( r=r_2 \)):

\[ \frac{\partial \hat{\rho}}{\partial r} = 0, \ -\frac{\partial \theta}{\partial z} = B_i z \cdot (\theta - \theta_c), \ \hat{u}_z = \hat{u}_r = 0 \] (5.13)

At the concentric annular ring (\( r=r_3 \)):

\[ \frac{\partial \hat{\rho}}{\partial r} = 0, \ -\frac{\partial \theta}{\partial z} = B_i z \cdot (\theta - \theta_c), \ \hat{u}_z = \hat{u}_r = 0 \] (5.14)
Initial conditions ($\tau=0$):
The metal hydride tank is initially assumed to be filled with hydrogen and to be at constant temperature equal to the reference temperature. In the following section the proposed model was used to perform dynamic optimization studies in order to compute the minimum release time of hydrogen.

5.3. Dynamic optimization of hydrogen desorption

From the discussion so far, it has been evident that there are certain parameters that must be optimally decided to achieve an economic and safe process operation. These parameters could be design specifications and manipulated variables of heat exchanger and bed reactor. The bed and heat exchanger design is assumed to be fixed and similar to the optimal design of Kikkinides et al. (2006b). Using this design, attention is placed on the optimal operation and control of the heating system. The overall objective is to minimize the total release-time of hydrogen from the metal-hydride bed reactor in order to achieve safe performance and optimality of the process – here, the operational/optimization variables to be determined correspond to quantities such as volumetric flow of the heating fluid and the corresponding inlet temperature.

In Papakosta (2008), a number of dynamic simulation studies showed that the rate of hydrogen release is approximately equal to the rate of increase of the temperature $T(z=1)$ at the exit of the reactor ($z=1$). Furthermore, it was illustrated that by increasing the heating medium flowrate $U_f$ the rate of hydrogen release decreases. Therefore the temperature $T(z=1)$ and the flowrate $U_f$ are considered as the controlled and manipulated variables for the control design studies that follow.

During the desorption process several design and operating constraints must be satisfied at all times (path constrains) to ensure a safe and economic operation, including heating fluid flowrate, pressure drop and bed temperature limitations.

\[
U_f \leq 3.87 \frac{m}{sec}, \quad T_{\text{max}} \leq 1.02 (296K), \quad \Delta p_f < 0.01 \text{atm}, \\
\Delta p_f = f \frac{\rho_f U_f^2}{2D_t} L, \quad f = 0.184 R e^{-0.2} D_t, \quad D_t = 2r_0 R
\] (5.15)

The desorption is completed when at least 99% of the total volume of hy-
drogen in the bed has been released which poses an additional constraint \( H_2(\%) > 0.99 \) at the end of the desorption process.

Dynamic optimization studies can be performed to obtain the optimal profiles of the manipulating and controlled variables for which the minimum time hydrogen release objective is achieved. The mathematical formulation of the dynamic optimization has the following general form:

\[
\min_{U_f, T, \dot{P}} t_s
\]

subject to:

1. Model equations (5.1)-(5.8)
2. Boundary conditions (5.9)-(5.14)
3. Path constrains (5.15)

where \( t_s \) is the time horizon of the desorption process. The optimization problem was developed in gPROMS Limited (2011). The centered finite difference (CFD) method was used to discretize the spatial and radial domain of the model using 30 elements.

Optimization results are presented in Figure 5.2 and 5.3 illustrating the optimal time profiles of the temperature \( T_{(z=1)} \) and hydrogen mass, respec-
Figure 5.3.: Optimal time profile of Hydrogen mass

tively. Figure 5.2 shows that the time needed for 99% desorption of the hydrogen mass is approximately 120 sec. The main idea is to maintain the reactor exit temperature $T_{(z=1)}$ close to the optimal profile shown in Figure 5.2, in order to achieve the minimum release time of hydrogen. In order to achieve this, especially in the presence of fluctuations of the operating conditions of the desorption process from the optimal ones, feedback control methods have to be employed. In this work we employ two feedback control methods the Proportional Integral (PI) control and multi-parametric MPC (mp-MPC) which will be discussed next. Based on the mathematical model presented above, a linear state space (SS) model approximation is derived, described next.

5.4. Model identification

A reduced order SS model is designed with model identification from the data of the simulations of the desorption process. The mathematical representation of the SS model without disturbances and with sampling time equal to 1 sec is as follows

$$x(t + 1) = Ax(t) + Bu(t)$$  \hspace{1cm} (5.16)
Figure 5.4.: Simulation and SS model comparison

\[ y(t) = Cx(t) + Du(t) \]  

(5.17)

where the system matrices are given as follows,

\[
A = \begin{bmatrix}
0.78 & 0.06 & -0.57 & -0.16 \\
0.04 & 0.97 & 0.09 & 0.11 \\
-0.03 & 0.02 & -0.05 & 0.82 \\
-0.41 & 0.13 & -0.36 & -0.09
\end{bmatrix}, 
B = \begin{bmatrix}
0.13 \\
-0.02 \\
0.14 \\
0.16
\end{bmatrix}, 
C = \begin{bmatrix}
-4.80 & 5.32 & 2.13 & 2.86
\end{bmatrix}, 
D = [0] \tag{5.18}
\]

A comparison between the reduced order SS model and the high-fidelity dynamic model is shown in Figure 5.4. The SS model closely approximates the behaviour of the process with a small approximation error (difference between the actual process output and the SS output) of 5\%. This approximation error is introduced as an uncertainty in the linear model, which has to be taken into account during the (robust) controller design.
5.5. Multi-Parametric Model Predictive Control (mp-MPC)

The next step involves the design of multi-parametric Model Predictive Controllers of the underlying metal-hydride bed. In model predictive control (MPC), an open-loop optimal control problem is solved at regular intervals (sampling instants), given the current process measurements, to obtain a sequence of the current and future control actions up to a certain time horizon (in a receding horizon control fashion), based on the future predictions of the outputs and/or states obtained by using a mathematical representation of the controlled system. Only the first input of the control sequence is applied to the system and the procedure is repeated at the next time instant when the new data are available. Being an online constrained optimization method, MPC not only provides the maximum output of the controlled process but also takes into account the various physical and operational constraints of the system.

The benefits of MPC have long been recognized from the viewpoint of cost and efficiency of operations. Nevertheless, its applications maybe restricted due to increased online computational requirements related to the constrained optimization. In order to overcome this drawback, explicit or multi-parametric model predictive control mp-MPC was developed (Pistikopoulos and Dua (2007b),Pistikopoulos and Dua (2007a)) which avoids the need for repetitive online optimization. In mp-MPC the online optimization problem is solved off-line with multi-parametric programming techniques to obtain the objective function and the control actions as functions of the measured state/outputs (parameters of the process) and the regions in the state/output space where these parameters are valid i.e. as a complete map of the parameters. The control is then applied by means of simple function evaluations instead of typically demanding online optimization computations.

In this step, two types of multi-parametric controllers are considered. First, a nominal mp-MPC controller is designed based on the linear SS model obtained in the previous section by considering A, B, C and D in (5.19) as constant matrices. Then a Robust mp-MPC controller is designed to explicitly account for the presence of model/parameter uncertainty, based on the method proposed by Pistikopoulos et al (2009) will be used (see
Appendix 1 for a brief description.

The following MPC formulation is considered for the hydrogen desorption process in the metal-hydride reactor

\[
\min_{u_{t+1}, \ldots, u_{t+N_u}} J = \sum_{i=1}^{N_y} Q (y_i - y_{ref,i})^2 + \sum_{j=0}^{N_u} R (u_j)^2
\]

\[\text{s.t. } x(t+1) = Ax(t) + Bu(t) \]
\[y(t) = Cx(t) + Du(t) \]
\[0 \leq u_i \leq 129, \ 1 \leq y_i \leq 1.2 \]
\[y_i = T_{(z=1)}(t+i), \ i = 1, \ldots, N_y \]
\[u_i = U_f(t+j), \ j = 0, \ldots, N_u - 1 \]
\[Q = 100, \ \rho = 0.001, \ R = 1 \]

(5.19)

where \(u_i\) are the manipulated variables, \(y_i\) are the controlled variable \(T_{(z=1)}\), \(y_{ref}\) is the optimal temperature profile, \(N_y\) is the total horizon and \(N_u\) the control horizon \((N_y = N_u = 5)\). The optimization problem involves five optimization variables \(u_{t+0}, \ldots, u_{t+4}\) and five parameters \(x=[x_1, x_2, x_3, x_4, y_{ref}]\) which represent the states at time zero and the reference point.

The objective function is set to minimize the quadratic norm of the error between \(T_{(z=1)}\) and its optimal profile while the constraints on \(u\) and \(y\) are also introduced.

For the case of constant system matrices, the optimization problem (5.19) is a multi-parametric Quadratic Programming (mp-QP) problem and can be solved with standard multi-parametric programming techniques Pistikopoulos and Morari (2002b). In our study, Parametric Optimization Software was used (ParOS Ltd, 2003) to obtain the explicit controller description, which is the optimal map of the control variables as function of the parameters of the system. This optimal map consists of 8 critical regions and the corresponding control laws. Each of the critical regions is described by a number of linear inequalities \(A_i x = b_i\) and its corresponding control is piecewise linear \(U_f = K_i x + c_i\), where \(i\) is the index of solutions. The critical regions are shown in Figure 5.5, based on a projection of the critical regions on the \(x_1 - x_2\) sub-spaces.

Then, a Robust MPC controller is designed by using the method presented in Appendix 1. The uncertainties \(\varepsilon_{\alpha}, \varepsilon_{\beta}\) are assumed to be equal to the approximation error from the identification step, which is 5% of the simulated model. The resulting 33 critical regions are shown in Figure 5.6, where it is
Figure 5.5.: Critical Regions of mp-MPC in the $x_1x_2$ sub-spaces

clearly shown that the parameter space has been reduced, compared to the corresponding space in Figure 5.5.

5.6. Simulation and validation of the multi-parametric controllers

The controllers are then implemented on the process (ie the detailed mathematical model) by applying at each sampling time the following explicit control law

$$\text{If } x \text{ in } A_i x = b_i \text{ Then } U_f = K_i x + c_i$$

For comparison purposes, a proportional-integral (PI) controller is also designed, with corresponding gains, $K_C$ and $\tau_I$

$$U_f(t) = U_{f0} + K_C \left( e(t) + \frac{1}{\tau_i} \int_0^t e(\tau) \, d\tau \right)$$

obtained by minimizing the ISE criterion

$$ISE = \int_0^t e^2(\tau) \, d\tau$$
Figure 5.6: Critical regions of mp-MPC in the $x_1 - x_2$ sub-spaces given by $K_C = -0.01$ and $\tau_i = 10$ sec.

Figure 5.7 depicts the simulation results of the mp-MPC, robust mp-MPC and PI implementation for nominal operating conditions (same conditions as in dynamic optimization), where $T_f = 350$K. It can be observed that all three controllers demonstrate similar behaviour, with the robust mp-MPC having the fastest response. Note that in the first few seconds the temperature diverges from the optimal temperature, but after 350 sec convergence to the optimal profile is achieved. The hydrogen release rate is shown in Figure 5.8, maintained close to the optimal one while in the first few seconds a small deviation is noticeable – all controllers achieve a 99% hydrogen release in 600sec.

In Figure 5.9, the simulation of both mp-MPC and PI controllers are shown when a perturbation is applied to the system, by decreasing the heating medium temperature to $T_f = 0.86 T_{f,\text{nom}}$ from its nominal value. For testing the performance of the controllers under hard process constraints, it is assumed that the temperature of the heating medium should not decrease below 0.9 (red line in Figure 5.9). Although the PI managed to reach the optimal profile at the end of the simulation without offset error, it violated the constraint at the beginning of the process. The explicit nominal mp-MPC controller had the fastest response from all three controllers, with a
Figure 5.7.: Temperature profiles of the mp-MPC, R mp-MPC and PI controllers - nominal conditions

Figure 5.8.: Hydrogen mass desorbed - nominal conditions
very small offset at the end, but it also violated the constraint even though it was closer to the feasible region than the PI. On the other hand, the only controller who managed to keep the system in the feasible region was the robust mp-MPC – constraint satisfaction was ensured at all times, albeit with a rather conservative performance, as noted by the resulting offset. This could be avoided by considering the uncertainty in the objective function.

5.7. Concluding Remarks

In this chapter an explicit MPC controller was applied in hydrogen desorption of metal-hydride bed storage tank. I developed dynamic model of a metal-hydride bed reactor for hydrogen storage based on previous works, which reformulated to capture more accurately the physical phenomena. Based on this model, dynamic optimization and control studies were performed to obtain the optimal temperature profile that can guarantee hydrogen release in minimum-time. A Robust mp-MPC controller was designed for the first time for metal hydride storages. The controller was compared to that of a PI controller and its performance was evaluated through simulations in several operating.
6. Modelling and explicit model predictive control for PEM fuel cell system

In this chapter we present an analytical dynamic model and a general framework for the optimal control design of a PEM fuel cell system of 2kW. The mathematical model consists of a detailed model for the PEM fuel cell stack and simplified models for the compressor, humidifier and cooling system. The framework features (i) a detailed dynamic process model, (ii) a reduced order approximating model obtained by performing dynamic simulations of the system and (iii) the design of an explicit/multi-parametric model predictive controller. The derived explicit/multi-parametric controller is tested and validated offline on several operating conditions. In chapter 7 we present the design of controllers for two systems, i) a PEM fuel cell system of 1.2kW Ballard with collaboration of the University of Seville, ii) a PEM fuel cell stack with collaboration of Chemical Process Engineering Research Institute.

6.1. PEM Fuel Cell System

Fuel cells are a promising technology for electrical power generation, widely regarded as a potential alternative for stationary and mobile applications. Fuel cells are electrochemical devices that convert the chemical energy of a fuel to electrical energy (Del Real et al. (2007)). The electrical efficiency of the fuel cells is higher than the most conventional devices for power generation, since they avoid intermediate steps of production of mechanical energy. The transport sector is one of the major contributors to global fossil fuel consumption and carbon emissions (Adamson, 2004). Alternative power devices for automotive applications have been actively studied over the last
few years with great emphasis on fuel cells (Dyer 2002). The primary type of fuel cells for automotive industry application is Proton Exchange Membrane (PEM) fuel cells, due to their suitable properties for vehicle applications such as low sensitivity to orientation favourable power to weight ratio and fast and easy start-up.

In order to use a fuel cell in an effective way, mathematical models are necessary to be able to analyse the system behaviour depending on the system design and operating conditions. The models developed in the literature can be classified into three main categories, namely, detailed fuel cell models based on partial differential equations (Fuller (1993), Dutta et al. (2001), Wang and Wang (2005)), steady-state fuel cell system based on experimental maps (Amphlett et al. (1995), Maher A.R. and Al-Janabi (2007)), and dynamic fuel cell system models that neglect spatial variations (Springer et al. (1991a), Ramousse et al. (2005)). Most of the publications on fuel cell modelling were developed at the cell level and included spatial variations of the fuel cell parameters. Membrane-electrode assembly (MEA) modelling is indeed the base of the entire PEM fuel cell system modelling (Pathapati et al. (2005), Shan and Choe (2005)). Complex electrochemical, thermodynamic and fluid dynamics principle were used to describe mathematically the entire physical environment of electrochemical reaction, the transport phenomena of gases, water, proton and electron, and as well as the relationships among fuel cell current, voltage, temperature, pressure, and materials (electrode, membrane and catalyst). The performance of the fuel cell under different steady-state operating conditions can be determined with those models. Some models are based on two-dimensional, steady-state fuel cell modelling (Golbert and Lewin (2004), Golbert and Lewin (2007)). More complex approaches in 3D modelling have also been developed (Maher A.R. and Al-Janabi (2007)), and the main purposes of those detailed models are to design the fuel cell components and to choose the fuel cell operating conditions, but they are computationally expensive and are not suitable for control studies. However, they establish the fundamental effects of operating parameters such as pressure and temperature on the fuel cell voltage.

Most of the models predict fuel cell polarization characteristics at different operating conditions. Amphlett et al. (1996) attempted to integrate together mechanistic model and empirical relation to derive advantages from
both of them. They developed a dynamic model for a PEM fuel cell stack which predicts fuel cell voltage and stack temperature for a given set of gas feed and operating conditions. Moreover, many publications addressed the water and thermal management of the fuel cell in order to describe the water, thermal, and reactant utilization of fuel cell by means of two or three-dimensional models (Fuller (1993)). Fronk (2000) demonstrated the importance of thermal management while trying to maximise the performance of fuel cell stack used within a vehicle. The system-level modelling is even more complicate than the MEA modelling and most of the models in the literature are based upon steady-state conditions. Steady-state models are typically used for component sizing, static trade-off analysis or cumulative fuel consumption (Pukrushpan (2005)). Despite the large number of developed models in the past 15 years (Bavarian et al. (2010)), there are still open issues regarding the development of models which are suitable for control and real-time estimation purposes. Current models are either too complex or they have not been sufficiently detailed to capture in detail the fuel cell dynamic behaviour. This field is in intense development, since such models are critical for future control development (Pukrushpan (2005), Del Real et al. (2007)). The concurrence of the current evolution in design of fuel cell systems and of the advanced integrated control techniques in microprocessor systems would allow the performance improvement of the portable fuel cell units (power regulation, heat management, water management etc.) (Pukrushpan (2005)). The recent literature has established that advanced control techniques have been deemed suitable for resolving the above issues (Bavarian et al. (2010)). This challenging control problem, to design a controller for an integrated fuel cell system as shown in figure 1.1.1, can be implemented mainly by using advanced control techniques, such as model based control. Model based control (MPC) strategy is a suitable approach to obtain the optimal operation of the fuel cell system, due to its ability to control multi-input multi-output systems with interactions and disturbances. However, MPC requires an analytical and accurate dynamic mathematical model of the system. The benefits of MPC have long been recognized from the viewpoint of cost and efficiency of operation. Nevertheless, its applications maybe restricted due to increased online computational requirements related to the constrained optimization. In order to overcome this drawback, explicit or multi-parametric model pre-
Predictive control mp-MPC was developed (Pistikopoulos and Dua (2007b), Pistikopoulos and Dua (2007a)) which avoids the need for repetitive online optimization. This chapter is focused on the mathematical modelling and control issues of PEM fuel cell systems according to the framework (Pistikopoulos (2009)) presented in figure 2, that comprises of the following steps.

1. Development of a high fidelity mathematical modelling - used for detailed simulation and (design and operational) optimization studies.

2. Development of a reduced order/approximating model, suitable for multi-parametric MPC

3. Design of multi-parametric MPC controllers

4. Off-line validation of the controllers

Step 1 involves the development of a high-fidelity mathematical model, for performing detailed dynamic simulation and design/operational optimization studies. The model is validated using experimental data in several operation conditions in order to guarantee the accuracy of the simulation results. In step 2, a reduced order approximated model is derived by performing system identification or reduced order techniques on the simulation data. Step 3 corresponds to the design of the multi-parametric/explicit Model Predictive Controllers (mp-MPC), by applying the available theory and tools of multi-parametric programming and control (Pistikopoulos and Dua (2007b), Pistikopoulos and Dua (2007a)). Finally step 4 involves the off-line validation of the derived multi-parametric/explicit controllers. In this chapter it is presented a systematic framework for the optimal control design of the PEM fuel cell system. A detailed mathematical model is first presented and dynamic simulations are performed based on which a reduced order state space (SS) model, suitable for the design of advanced model based controllers for temperature and voltage regulation, is derived. Finally, the controller introduced in the process model and its performance is validated.
6.2. Types of fuel cells

There are different types of fuel cells and they could be grouped by the type of electrolyte they use Carrette et al. (2001); O’Hayre et al. (2009),

- phosphoric acid fuel cells (PAFC)
- polymer electrolyte membrane fuel cells (PEMFC)
- alkaline fuel cells (AFC)
- molten carbon fuel cells (MCFC)
- solid-oxide fuel cells (SOFC)

Table 6.1 gives an overview of the different types and includes their operation temperature, the most common power outputs of the cell type and the possible applications.

Polymer electrolyte membrane fuel cells are working at ambient temperature and pressure and have a high power density (Mehta and Smith Cooper, 2003), which allow PEM fuel cells to be used in a number of applications, such as in transport sector (Mehta and Smith Cooper, 2003). Moreover, they can be used in portable devices with the main advantage over the batteries of no charging times (Dyer, 2002) and also can reduce the costs for energy by up to 80% (Chu et al., 2001). Tueber et al. (2003) developed an external fuel cell system able to power a portable computer in 2002.

The PEM fuel cells can also be used in large scale industrial applications for utilizing the by-products of a chemical process and produce electricity. The chemical company Solvay together with the fuel cell producer NedStack are planning to build a 1 MW hydrogen fuel cell power plant. It shall supply the power for a chlor-alkali plant in Belgium and will be using hydrogen which is a by-product of the chlorine production on the site. Additionally to producing power from a by-product of the plant the aim of the project partners is to demonstrate the economic feasibility and dynamics of such a plant which would be the largest ever build PEM fuel cell in the world as well as improve the cell’s performance regarding its lifetime and costs.
<table>
<thead>
<tr>
<th></th>
<th>AFC</th>
<th>PEMFC</th>
<th>DMFC</th>
<th>PAFC</th>
<th>MCFC</th>
<th>SOFC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>alkaline</td>
<td>polymer electrolyte membrane</td>
<td>direct methanol</td>
<td>phosphoric acid</td>
<td>molten carbonate</td>
<td>solid oxide</td>
</tr>
<tr>
<td>T</td>
<td>&lt; 100°C</td>
<td>60-120°C</td>
<td>60-120°C</td>
<td>160-220°C</td>
<td>600-800°C</td>
<td>800-1000°C</td>
</tr>
<tr>
<td>Power</td>
<td>5-150 kW</td>
<td>0.1-250 kW</td>
<td>5 kW</td>
<td>50 kW - 11 MW</td>
<td>100 kW - 2 MW</td>
<td>0.1-250 kW</td>
</tr>
<tr>
<td></td>
<td>small plants</td>
<td>small plants</td>
<td>small plants</td>
<td>small/medium plants</td>
<td>small power plants</td>
<td>small plants</td>
</tr>
<tr>
<td>Charge carrier in the electrolyte</td>
<td>OH⁻</td>
<td>H⁺</td>
<td>H⁺</td>
<td>H⁺</td>
<td>CO³⁻</td>
<td>O²⁻</td>
</tr>
<tr>
<td>Application</td>
<td>portable applications</td>
<td>Dyer (2002)</td>
<td>Transportation</td>
<td>Space</td>
<td>Military</td>
<td>combined heat &amp; power for decentralised systems and for transportation (trains, boats...)</td>
</tr>
</tbody>
</table>

Table 6.1: Types of fuel cells Carrette et al. (2001)
6.3. Basic Operation of a PEM fuel cell

The electrochemical reactions in fuel cell happen simultaneously on both sides of the membrane (anode, cathode). The underlying reaction for the energy conversion is the oxidation of hydrogen. In principle this is the same reaction as in a combustion engine fuelled with hydrogen and is an exothermic reaction, which means that there is energy released in the process.

\[ 2 \text{H}_2 + \text{O}_2 \rightleftharpoons 2 \text{H}_2\text{O} \]  \hspace{1cm} (6.1)

This is a redox reaction, since hydrogen is oxidised and oxygen reduced. At the anode the hydrogen fuel is oxidised resulting in the formation of a proton and an electron.

\[ \text{H}_2 \rightleftharpoons 2\text{H}^+ + 2\text{e}^- \]  \hspace{1cm} (6.2)

The oxygen reduction takes place at the cathode, yielding water as product. This is removed from the cell by an excess flow of of the reactant gases.

\[ \frac{1}{2} \text{O}_2 + 2\text{H}^+ + 2\text{e}^- \rightleftharpoons \text{H}_2\text{O} \]  \hspace{1cm} (6.3)

In order for the reactions to proceed continuously the electrons and protons involved in both reactions have to pass from the anode to the cathode. A membrane within the fuel cell allows only the transportation of the protons while the electrons pass through the external circuit. The major steps involved in the generation of electricity in a PEM fuel cell are (O’Hayre et al., 2009):

- the delivery of reactants to the reaction sites
- the electrochemical reaction
- the conduction of ions and electrons from the anode to the cathode
- the product removal

6.4. PEM Fuel Cell Mathematical Model

The PEM fuel system model is described analytically in this section. The system under consideration is a PEM fuel cell stack, a compressor and a
cooling system to maintain the temperature of the stack (Figure 11.1). Hydrogen is channelled in the anode side of the fuel cell while air in the cathode side. The compressor and the electric drive motor are used to achieve the desired air massflow and pressure, while the humidifier has been used to achieve proper humidity of the air in order to minimize the danger of dehydration of the membrane. In addition, a recycling system for hydrogen is applied to minimize the hydrogen consumption. The proposed model involves three main modules: fluid dynamics model, thermodynamic model and electrochemical static model. The theoretical equations are combined with experimental/empirical formulations, resulting a semi-empirical dynamical model. Fluid dynamics model is composed of three interconnected modules; anode and cathode flow stream, and the membrane. The thermodynamic model is used to determine the homogeneous temperature of the fuel cell and to design the appropriate cooling system, while electrochemical static model, described in the following section, used to predict stack voltage. The model assumptions are discussed as the equations are presented and a summary of those are presented below,

1. Ideal gas law was employed for gaseous species
2. Laminar flow of the gases
3. Isotropic and homogeneous electrolyte, electrode, and bipolar material structures
4. Negligible ohmic potential drop in components
5. Mass and energy transport is modelled from a macroperspective using volume-averaged conservation equations
6. The product water generated at the cathode is assumed to be in the liquid state
7. Uniform stack temperature of the fuel cell due to high thermal conductivity
8. Uniform pressure of the stack and anode, cathode
9. Constant thermophysical properties
10. Outlet valve of the cathode allows isentropic expansion
11. Purge valve removes all the liquid water

The dynamic mathematical model (0D), developed in this work, includes mass balances for the anode and cathode side and recirculation, semi-empirical equations for the membrane, electrochemical equations, heat balances for the fuel cell and mass and energy equation for the humidifier, compressor and the cooling system.
## Nomenclature

<table>
<thead>
<tr>
<th>Nomenclature</th>
<th>Superscript or subscript</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>activity [-]</td>
</tr>
<tr>
<td>$A$</td>
<td>area [mÅs]</td>
</tr>
<tr>
<td>$c$</td>
<td>concentration [mol.cm$^{-3}$]</td>
</tr>
<tr>
<td>$c_p$</td>
<td>specific heat capacity [J.kg$^{-1}$.K$^{-1}$]</td>
</tr>
<tr>
<td>$D_w$</td>
<td>diffusion coefficient of water in the membrane [cm$^2$.s$^{-1}$]</td>
</tr>
<tr>
<td>$E$</td>
<td>potential [V]</td>
</tr>
<tr>
<td>$F$</td>
<td>Faraday constant [C.mol$^{-1}$]</td>
</tr>
<tr>
<td>$I$</td>
<td>current [A]</td>
</tr>
<tr>
<td>$J$</td>
<td>inertia [kg.mÅs]</td>
</tr>
<tr>
<td>$K$</td>
<td>valve coefficient [kg.bar$^{-1}$.s$^{-1}$]</td>
</tr>
<tr>
<td>$m$</td>
<td>mass [kg]</td>
</tr>
<tr>
<td>$M$</td>
<td>molar mass [kg.mol$^{-1}$]</td>
</tr>
<tr>
<td>$n$</td>
<td>molar quantity [mol]</td>
</tr>
<tr>
<td>$N$</td>
<td>net water flow [mol.s$^{-1}$.cm$^{-2}$]</td>
</tr>
<tr>
<td>$nd$</td>
<td>electro-osmotic drag coeffi- cient [-]</td>
</tr>
<tr>
<td>$P$</td>
<td>pressure [Pa] or power [W]</td>
</tr>
<tr>
<td>$Q$</td>
<td>heat flow [W]</td>
</tr>
<tr>
<td>$R$</td>
<td>resistance [W]</td>
</tr>
<tr>
<td>$R$</td>
<td>ideal gas constant [J.mol$^{-1}$.K$^{-1}$]</td>
</tr>
<tr>
<td>$RH$</td>
<td>relative humidity [-]</td>
</tr>
<tr>
<td>$t$</td>
<td>time [s]</td>
</tr>
<tr>
<td>$T$</td>
<td>temperature [K]</td>
</tr>
<tr>
<td>$U$</td>
<td>heat transfer coefficient [W.m$^{-2}$.K$^{-1}$]</td>
</tr>
<tr>
<td>$V$</td>
<td>volume [m$^3$] or voltage [V]</td>
</tr>
<tr>
<td>$w$</td>
<td>humidity ratio [-]</td>
</tr>
<tr>
<td>$x$</td>
<td>molar fraction [-]</td>
</tr>
<tr>
<td>$y$</td>
<td>mass fraction [-]</td>
</tr>
</tbody>
</table>

- $a_0$: reference condition
- $amb$: ambient
- $an$: anode
- $atm$: atmosphere
- $ca$: cathode
- $chem$: chemical
- $cool$: coolant
- $comp$: compressor
- $dry$: dry
- $elec$: electrical
- $evap$: evaporation
- $fc$: fuel cell
- $gen$: generated
- $hyd$: hydraulic
- $im$: inlet manifold
- $membr$: membrane
- $ohm$: ohmic
- $om$: outlet manifold
- $out$: outlet
- $purge$: purge
- $rad$: radiation
- $react$: reacted
- $recirc$: recirculation
- $sat$: saturation
- $st$: stack
- $tank$: tank
- $vap$: vapour
- $volumetric$: volumetric
6.4.1. Cathode Mathematical Model

The cathode mathematical model describes the air flow behaviour in the cathode side of the fuel cell. The equations are based on the first principle equations and include the mass balances for the components (Pukrushpan (2005)). The inlet mass flow rates of the three elements, namely oxygen, nitrogen and vapour can be calculated using the humidity ratio. Considering a mixture of air and water vapour, the humidity ratio is defined as the ratio of the mass of water vapour to the mass of dry air (Del Real et al. (2007)).

\[
\dot{m}_{v,ca,in} = \frac{M_{v,H_2O}}{M_{air}} \frac{\phi_{ca,in} p_{sat}(T_{ca,in})}{p_{ca,in} - \phi_{ca,in} p_{sat}(T_{ca,in})}
\]

where \(\phi_{ca,in}\)is the relative humidity and \(p_{sat}\), the pressure of vapour saturation, \(M_i\) (i is O2,N2,dry air) is the molar mass of the component \(T_{ca,in}\) the inlet temperature (Pukrushpan (2005)).

\[
\dot{m}_{O_2,ca,in} = x_{O_2,ca,in} \frac{1}{1 + w_{v,ca,in}} \dot{m}_{ca,in}
\]

\[
\dot{m}_{N_2,ca,in} = x_{N_2,ca,in} \frac{1}{1 + w_{v,ca,in}} \dot{m}_{ca,in}
\]

\[
\dot{m}_{v,ca,in} = 1 - \frac{1}{1 + w_{v,ca,in}} \dot{m}_{ca,in}
\]
\[ x_{O_2,ca,in} = \frac{m_{O_2,ca,in}}{m_{dryair}} = \frac{y_{O_2,ca,in} M_{O_2}}{y_{O_2,ca,in} M_{O_2} + (1 - y_{O_2,ca,in}) M_{N_2}} \]  

(6.8)

where \( m_{i,ca} \) is the mass flowrate in cathode side (i is \( O_2, N_2, \) vapour, dry air), \( x_{i,ca,in} \) is molar fraction in the inlet of channel of cathode side (i is \( O_2, N_2, \)) and \( y_{O_2,ca,in} \) is the oxygen molar fraction. The mass continuity equation is used to balance the mass of the elements inside the cathode Pukrushpan (2005)

\[ \dot{m}_{O_2,ca} = \dot{m}_{O_2,ca,in} - \dot{m}_{O_2,ca,out} - \dot{m}_{O_2,react} \]  

(6.9)

\[ \dot{m}_{N_2,ca} = \dot{m}_{N_2,ca,in} - \dot{m}_{N_2,ca,out} \]  

(6.10)

\[ \dot{m}_{v,ca} = \dot{m}_{v,ca,in} - \dot{m}_{v,ca,out} + \dot{m}_{v,membr} + \dot{m}_{evap,ca} \]  

(6.11)

where \( \dot{m}_{O_2,react} \) is the reacted oxygen, \( \dot{m}_{v,membr} \) is the water mass flowrate across the membrane, \( \dot{m}_{evap} \) is the evaporation mass and \( \dot{m}_{H_2,react} \) is reacted mass of hydrogen. It is assumed that no liquid water carried by the inlet air enters the cathode channel according to the suggestion of Del Real et al. (2007), thus \( m_{l, in} = 0 \). The water produced from the reaction assumed to be in liquid form and has to be evaporated, partially or totally depending on the cathode operating conditions. Consequently, the mass balance of the liquid can be described by the following equation.

\[ \dot{m}_{l,ca} = \dot{m}_{l,ca,gen} - \dot{m}_{evap,ca} \]  

(6.12)

where \( \dot{m}_{l,ca,gen} \) is the generated mass of liquid water. The dynamic equation for the evaporation inside the cathode channel can be described by the following equation

\[ \dot{m}_{evap,ca} = \min \left( A_{fc} \left( p_{sat} (T_{st}) - p_{v,ca} \right) \sqrt{\frac{M_v}{2\pi R T_{st}}}, \dot{m}_{l,ca,gen} \right) \]  

(6.13)

where \( A_{fc} \) is the active area, \( R \) is the ideal gas constant and \( p_{v,ca} \) the vapour pressure in the cathode channel. When the stream pressure is smaller than the saturation pressure, the water is evaporated and the value of \( (p_{sat} (T_{st}) - p_{v,ca}) \) is positive, while water condenses when the value is negative. As it has been assumed, that no liquid water is brought by the air inlet, it is not possible to evaporate more water than produced by the chemical reaction. Using the ideal gas law, the partial pressure of oxygen,
nitrogen and water vapour inside the cathode channel has been calculated as follows

\[ p_{i,ca} = \frac{R_i T_{st}}{V_k} m_{i,ca} \text{ with } i = O_2, N_2, v \]  

(6.14)

where \( p_{i,ca} \) is the partial pressure and \( R_i \) the gas constant of each component (i is O2, N2, vapour pressure), and \( V_k \) is the volume of the cathode side. The total cathode pressure can be calculated by adding the partial pressures using the Dalton law.

\[ p_{ca} = \sum p_{i,ca} \]  

(6.15)

Electrochemical principles are used to calculate the rate of oxygen consumption and water production in the fuel cell, and they are functions of the stack current (Ist) and the number of cells (Nfc).

\[ \dot{m}_{O_2,\text{react}} = \frac{N_f c M_{O_2} I}{4F} \]  

(6.16)

\[ \dot{m}_{i,ca,\text{gen}} = \frac{N_f c M_i I}{2F} \]  

(6.17)

where F is the Faraday constant. The outlet massflow rates for oxygen, nitrogen and water vapour can be calculated from the equations (15-18). The equation (6.18) assumes an isentropic expansion and considers the valve as a nozzle with a convergent and a divergent section. The section \( A_t \) (m2), represents the opening of the valve and will be controlled to regulate the pressure inside the cathode side of the fuel cell (Thomas (1999)).

\[ \dot{m}_{ca,\text{out}} = A_t \sqrt{\frac{2}{\gamma - 1} \frac{p_{ca}}{v_{air}}} \left[ \left( \frac{p_{out}}{p_{ca}} \right)^{\frac{\gamma}{2}} - \left( \frac{p_{out}}{p_{ca}} \right)^{\frac{\gamma + 1}{2}} \right] \]  

(6.18)

\[ \dot{m}_{O_2,ca,\text{out}} = \frac{m_{O_2,ca}}{m_{O_2,ca} + m_{N_2,ca} + m_{v,ca}} \dot{m}_{ca,\text{out}} \]  

(6.19)

\[ \dot{m}_{N_2,ca,\text{out}} = \frac{m_{N_2,ca}}{m_{N_2,ca} + m_{O_2,ca} + m_{v,an}} \dot{m}_{ca,\text{out}} \]  

(6.20)

\[ \dot{m}_{v,ca,\text{out}} = \frac{m_{v,ca}}{m_{O_2,ca} + m_{N_2,ca} + m_{v,ca}} \dot{m}_{ca,\text{out}} \]  

(6.21)

where \( P_{out} \) is the outlet pressure, \( v_{air} \) is the air specific volume (m3.kg-1) and \( \gamma \) is the heat capacity ratio.
6.4.2. Anode Mathematical Model

The fuel cell system under consideration assumes storage of hydrogen in metal hydride bed reactor. The optimal operation of the storage tank has significantly role on the operation of fuel cell since the pressure and the temperature of the tank have to be regulated in order to achieve the demanded hydrogen mass flowrate. Since the scope of this work is not to investigate the modelling and control of the metal hydride tank we will focus on the fuel cell system. However, further reading on this topic can be found on Kikkinides et al. (2006a) for the modelling and optimization. The difference between the pressure of the anode and cathode channel should be regulated, hence a valve is needed to adjust the pressure of the fuel and drops it to close to cathode channel. Similar to cathode model, a dynamic model for the anode channel is described in this section. This model includes dynamic mass balances, partial pressure equations, outlet flow of the anode side and the recirculation of the unreacted hydrogen. The equations that model the anode flow are analogous to the ones that model the cathode flow (Pukrushpan (2005)). The mass balance is then expressed as follows

\[ \dot{m}_{H_2,an} = \dot{m}_{H_2,an,in} - \dot{m}_{H_2,an,out} - \dot{m}_{H_2,reacted} \] (6.22)

\[ \dot{m}_{v,an} = \dot{m}_{v,an,in} - \dot{m}_{v,an,out} - \dot{m}_{v,memb} + \dot{m}_{evap,an} \] (6.23)

\[ \dot{m}_{l,an} = \dot{m}_{l,an,in} - \dot{m}_{l,an,out} - \dot{m}_{evap,an} \] (6.24)

where \( m_{i,an} \) is the mass flowrate in anode side (i is \( H_2, \text{vapour,liquid} \)), \( m_{H_2,react} \) is the reacted hydrogen, \( m_{v,memb} \) is the water mass flowrate across the membrane, \( m_{evap,an} \) is the evaporation mass. The assumption of no liquid water is considered at the inlet of the anode channel which leads to the conclusion that no liquid water is considered after the recirculation loop using a water separator just before the inlet and that the hydrogen coming from the tank is supposed to be dry (\( m_{l,an,in} = 0 \)). Moreover, it is assumed that all the liquid water has been removed by the purge valve through the following equation

\[ \dot{m}_{l,an,out} = \frac{m_{l,an}}{t_{purge}} \] (6.25)
Equations for humidity ratio, inlet flow rates and the evaporation mass of the anode channel are similar to the cathode.

\[
w_{v,an,in} = \frac{M_{v,H_2O}}{M_{H_2}} \frac{\phi_{an,in} p_{sat}(T_{an,in})}{p_{an,in} - \phi_{an,in} p_{sat}(T_{an,in})} \tag{6.26}
\]

\[
\dot{m}_{H_2,an,in} = \frac{1}{1 + w_{v,an,in}} \dot{m}_{an,in} \tag{6.27}
\]

\[
\dot{m}_{v,an,in} = \frac{1}{1 + w_{v,an,in}} \dot{m}_{an,in} \tag{6.28}
\]

\[
\dot{m}_{evap,an} = \min \left( A_{fc} (p_{sat}(T_{st}) - p_{v,anch}) \sqrt{\frac{M_v}{2\pi RT_{st}}}, 0 \right) \tag{6.29}
\]

where \( p_{v,an} \) the vapour pressure of the anode channel. Given that no liquid water enters the anode channel and the water coming from the cathode through the membrane is only vapour, evaporation inside the anode has been limited, allowing only condensation. The partial pressures inside the channel are also obtained using the ideal gas law

\[
p_{i,an} = \frac{R T_{st}}{V_k} m_{i,an} \text{with } i = H_2, v \tag{6.30}
\]

and the Dalton law

\[
p_{an} = \sum p_{i,an} \tag{6.31}
\]

The amount of hydrogen reacted inside the fuel cell is a function of the stuck current and the number of cells.

\[
\dot{m}_{H_2,react} = \frac{N_{fc} M_{H_2} I}{2F} \tag{6.32}
\]

### 6.4.3. Anode Recirculation

There are not many descriptions of the recirculation loop (Figure 3) for the hydrogen in the open literature (Bao et al. (2006)), and the equations have been suggested trying to best fit to the actual behaviour in the actual fuel cell. The inlet flow of the anode is obtained by mixing the flow coming from the recirculation and the flow of dry hydrogen coming from the tank inside the T. The pressure at the inlet is calculated using the ideal gas law as follows
where \( P_i \) is the pressure, \( n_i \) is the moles and \( T_i \) the temperature of the stream \( i \) (\( i \) is anode inlet, dry hydrogen and recirculation stream), The pressure inside the recirculation system has to be higher than the pressure at the inlet, in order to avoid counter flow inside the recirculation system. The anode inlet pressure has to be controlled considering cathode channel pressure in order to avoid different pressures of the two inlet streams. Between the tank of dry hydrogen and the inlet anode channel there is a valve and the mass flowrate coming from the tank can be calculated as follows (Thomas (1999))

\[
\dot{m}_{\text{dry},H_2} = A \sqrt{\frac{2 \gamma}{\gamma - 1}} \frac{p_{\text{tank}}}{v_{H_2}} \left[ \frac{p_{\text{dry},H_2}^{\frac{2}{\gamma}}}{p_{\text{tank}}} - \left( \frac{p_{\text{dry},H_2}}{p_{\text{tank}}} \right)^{\frac{\gamma+1}{\gamma}} \right] \quad (6.34)
\]

Consequently, the inlet mass flowrate in the anode channel can be calculated by the sum of the mass flow rate of the recirculation stream and the hydrogen coming from the metal hydride tank.

\[
\dot{m}_{\text{an,in}} = \dot{m}_{\text{dry},H_2} + \dot{m}_{\text{recirc},H_2} \quad (6.35)
\]

The temperature of this inlet stream is calculated considering a mixing of both flows.

\[
\int_{T_1}^{T} \dot{m}_1 c_p_1 dt + \int_{T_2}^{T} \dot{m}_2 c_p_2 dt = 0 \quad (6.36)
\]

which results to the following equation

\[
T_{\text{an,in}} = \frac{\dot{m}_{\text{dry},H_2} c_p_{\text{dry},H_2} T_{\text{dry},H_2} + \dot{m}_{\text{recirc},H_2} c_p_{\text{recirc},H_2} T_{\text{recirc},H_2}}{\dot{m}_{\text{dry},H_2} c_p_{\text{dry},H_2} + \dot{m}_{\text{recirc},H_2} c_p_{\text{recirc},H_2}} \quad (6.37)
\]
6.4.4. Fuel cell outlet

The anode outlet flowrate $m_{an,out}$ represents the purge of anode side in order to remove accumulated gases and liquid water for the anode side. In the designed PEM fuel cell system assumed purge equal to 0.5 but could vary within the range of zero and one, depending on the specifications. With the knowledge of the outlet flowrate of the anode channel, the mass flowrate of the hydrogen and vapour could be calculated similar to the inlet flow.

\[
\dot{m}_{H_2,out} = \frac{m_{H_2,an}}{m_{H_2,an} + m_{v,an}} \dot{m}_{an,out} \quad (6.38)
\]

\[
\dot{m}_{v,out} = \frac{m_{v,an}}{m_{H_2,an} + m_{v,an}} \dot{m}_{an,out} \quad (6.39)
\]

The outlet flowrate of the stack is determined using an outlet manifold equation

\[
\dot{m}_{an,out} = K_{an}(P_{an} - P_{om}) \quad (6.40)
\]

where $K_{an}$ is the outlet manifold coefficient, and $P_{om}$ is the anode outlet manifold pressure.

6.4.5. Membrane Hydration Model

The hydration model determines the rate of mass flowrate across the membrane and the water content of the membrane. Semi-empirical equations have been proposed to describe the phenomena with different constant coefficients. The water transport across the membrane is achieved through two distinct phenomena. Water electro-osmotic drag and back diffusion from the cathode contribute to water transport in the membrane. The former occurs when protons migrate through the membrane from the anode to the cathode and carry water molecules with them. The water drag from the anode to the cathode runs proportionally to the proton flow and thus this phenomenon increases at higher-current density. The latter originates from the diffusion of water through the membrane, driven by the water-concentration gradient in the thickness profile. The phenomenon of water back diffusion across a membrane from the cathode to the anode usually predominates owing to the water produced at the cathode. The combination of those two phenomena results to the following equation, which determines the water flow across
the membrane

\[ N_{v,\text{memb}} = nd \frac{I}{A_{fc} F} - D_w \frac{c_{v,ca} - c_{v,an}}{\delta_{\text{memb}}} \]  (6.41)

where I is the stuck current, Dw is the diffusion coefficient of the water in membrane, \( \delta_{\text{memb}} \) is the thickness of the membrane, \( c_{v,ca} \) and \( c_{v,an} \) is the water concentration in cathode and anode channel respectively. The total stack mass flow rate across the membrane can be computed by the following equation

\[ \dot{m}_{v,\text{memb}} = N_{v,\text{memb}} M_v A_{fc} N_{fc} \]  (6.42)

where \( M_v \) is the vapour molar mass, \( N_{fc} \) is the number of fuel cells and \( A_{fc} \) is the fuel cell active area. The water concentration at the membrane surfaces on anode and cathode sides depends on the membrane water content \( \lambda_k \) and can be obtained using the following equation

\[ c_{v,k} = \frac{\rho_{\text{mem,dry}}}{M_{\text{memb,dry}}} \lambda_k \]  (6.43)

where \( \rho_{\text{mem,dry}} \) is the membrane dry density, \( M_{\text{memb,dry}} \) is the membrane dry equivalent weight and \( k \) refers to anode or cathode side. The water content in the membrane \( \lambda_k \) is defined as the ratio of water molecules to the number of charge sites. These equations are developed based on experimental results (Pukrushpan (2005), Del Real et al. (2007)).

\[ \lambda_k = \begin{cases} 0.043 + 17.81a_k - 39.85a_k^2 + 36.0a_k^3, & 0 < a_k \leq 1 \\ 14 + 1.4(a_k - 1), & 1 < a_k \leq 3 \end{cases} \]  (6.44)

where \( a_i \) is the water activity. The membrane water content can be calculated using the activities of the gas in cathode and anode side and the average water activity between anode and cathode water activities is calculated below.

\[ a_k = \frac{p_{v,k}}{p_{s\text{at},k}}, a_{\text{memb}} = \frac{a_{an} + a_{ca}}{2} \]  (6.45)

The electro-osmotic drag coefficient \( nd \) and the water diffusion coefficient \( Dw \) is then calculated using the membrane water content (McKay et al. 2012).
2005 McKay et al. (2005), Dutta et al. (2001)).

\[
D_w = D_{\lambda_{an}} \exp\left(2416 \left(\frac{1}{303} - \frac{1}{T_{st}}\right)\right)
\]  

(6.46)

where

\[
D_{\lambda_{an}} = \begin{cases} 
10^{-6} & \lambda_{an} < 2 \\
10^{-6} (1 + 2 (\lambda_{an} - 2)) & 2 \leq \lambda_{an} \leq 3 \\
10^{-6} (3 - 1.67 (\lambda_{an} - 3)) & 3 < \lambda_{an} < 4.5 \\
1.25 \times 10^{-6} & \lambda_{an} \geq 4.5 
\end{cases}
\]  

(6.47)

and

\[
n_d = 0.0029 \lambda_{an}^2 + 0.05 \lambda_{an} - 3.4 \times 10^{-19}
\]  

(6.48)

where T_{st} is the stack temperature.

### 6.4.6. Electrochemistry

In this section the steady state mathematical modelling of the fuel cell voltage is explained. The voltage is a function of stack current, hydrogen and oxygen partial pressure, fuel cell temperature and membrane humidity ratio. Nerst equation has been used to calculate the stack voltage, which include the Nerst potential, activation losses, ohmic losses and concentration losses.

\[
V_{st} = N_{fc} (E_{Nernst} - V_{act} - V_{ohm} - V_{conc})
\]  

(6.49)

\[
E_{nernst} = 1.23 - 8.5 \times 10^{-4} (T - 298) + \frac{RT}{2F} [\ln(p_{H_2}) + 0.5 \ln(p_{O_2})]
\]  

(6.50)

Activation loss or activation overvoltage occurs because of the need to move electrons and to break chemical bonds in the anode and cathode. Part of available energy is lost in driving chemical reactions that transfers electrons to and from electrodes. However, the oxidation of H2 at the anode is very rapid and the reduction of O2 at the cathode is slow, then the voltage drop due to activation loss is dominated by the cathode reaction.

\[
V_{act} = \xi_1 + \xi_2 T + \xi_3 T \ln(I) + \xi_4 T \ln(C_{O_2}^*)
\]  

, with, 

\[
C_{O_2}^* = \frac{P_{O_{2,ca}}}{5.08 \times 10^6 \exp\left(-\frac{498}{T_{st}}\right)}
\]  

(6.51)

126
Ohmic loss arises from resistance of polymer membrane to the transfer of proton and the resistance of the electrode and collector plate to the transfer of electron. Resistance depends on the membrane humidity and cell temperature.

\[ V_{ohm} = r_{int}I = (\xi_5 + \xi_6T + \xi_7I)I \]  

(6.52)

Finally concentration loss results from the drop in concentration of the reactants as they are consumed in the reaction. This explains the rapid voltage drop at high current density.

\[ V_{conc} = -\frac{RT}{2F} \ln\left(1 - \frac{I}{I_{lim}}\right) \]  

(6.53)

The calculation of the stack voltage includes semi-empirical equations, thus many authors have found different values for the coefficients \( \xi \) (Amphlett et al. (1996), Pathapati et al. (2005), Park and Choe (2008)). The gross power produced by the fuel cell is also obtained as follows

\[ P_{st} = V_{st}I_{st} \]  

(6.54)

To obtain the net power, the power consumed by the auxiliary equipments has to by remove.

\[ P_{net} = P_{st} - P_{aux} \]  

(6.55)

6.4.7. Thermodynamic balance

The temperature significantly affects the performance if the fuel cell by influencing the water removal and reactants activity. Heat is generated by the operation of the fuel cell since the enthalpy that is not converted to electrical energy will instead be converted to thermal energy. Thus the temperature will rise beyond the operating temperature range of the fuel cell and must be kept in the appropriate range using a cooling system. Two main assumptions are introduced in the heat balances, (i) any fuel energy that is not converted into electrical energy is converted into heat. (Lee (1998)) and (ii) the temperature at the anode and cathode side are equal to the fuel cell stack temperature (uniform temperature). The energy model is using the main terms of the overall energy balance using the following
principle

\[
\begin{pmatrix}
\text{Energy Accumulation} \\
\end{pmatrix} = 
\begin{pmatrix}
\text{Inlet Energy} \\
\end{pmatrix} - 
\begin{pmatrix}
\text{Outlet Energy} \\
\end{pmatrix} + 
\begin{pmatrix}
\text{Generated Energy} \\
\end{pmatrix}
\]

Mathematical model for fuel cell system with energy balances has been presented by Del Real et al. (2007), as well as by MÁijller et al. (2006) Muller (2006). The mathematical model for radiation has been suggested by Wishart et al. (2006) while the latent heat has been calculated as follows.

\[ m_{st}C_{p_{st}} \frac{dT_{st}}{dt} = \dot{Q}_{in} - \dot{Q}_{out} - \dot{Q}_{chem} - \dot{Q}_{elec} - \dot{Q}_{cool} - \dot{Q}_{rad} - \dot{Q}_{latent} \]  (6.56)

The inlet and outlet heat flowrate by the chemicals elements entering the fuel cell can be calculated by the following equations

\[ \dot{Q}_{in} = \sum_i \dot{m}_{i,in}C_{p_i}T_{in} \]  (6.57)

\[ \dot{Q}_{out} = \sum_i \dot{m}_{i,out}C_{p_i}T_{st} \]  (6.58)

The generated energy by chemical reactions (\(\dot{Q}_{chem}\)), radiation energy(\(\dot{Q}_{rad}\)), the latent energy(\(\dot{Q}_{latent}\)) due to change of water phase and the heat flowrate in the form of electricity(\(\dot{Q}_{elec}\)) in the PEM fuel cell stack can be calculated as follows

\[ \dot{Q}_{chem} = \dot{m}_{H_2Ogen}\Delta_r H^o(T^0) + C_{PH_2O}(T_{st} - T^0) \]

\[ -\dot{m}_{O_2react}C_{PO_2}(T_{in} - T^0) - \dot{m}_{H_2react}C_{PH_2}(T_{in} - T^0) \]  (6.59)

\[ \dot{Q}_{elec} = V_{st}I_{st} \]  (6.60)

\[ \dot{Q}_{rad} = \varepsilon\sigma A_{rad}(T_{st}^4 - T_{amb}^4) \]  (6.61)

\[ \dot{Q}_{latent} = M_{H_2O} \dot{m}_{evap} H_{vaporization} \]  (6.62)

\[ H_{vaporization} = 45070 - 41.9T_{st} + 3.44 \times 10^{-3}T_{st}^2 + 2.54 \times 10^{-6}T_{st}^3 - 8.98 \times 10^{-10}T_{st}^4 \]  (6.63)

where \(\Delta H_{ro}\) is the mass specific enthalpy of formation of liquid water, \(C_{PH2O}, C_{PO2}, C_{PH2}\) is the thermal capacity of water, oxygen and hydrogen respectively, \(\varepsilon\) is the emissivity, \(\sigma\) is the Stefan-Boltmann constant, \(A_{rad}\) is the radiation exchange area, \(H_{vaporization}\) is the vaporization enthalpy,
Tamb is the ambient temperature, Tin is the inlet temperature of the reactants and T0 is the reference temperature (298K). Simulation showed that radiation and latent heat flows are negligible compared to the others. The energy model considering the fuel cell as a CSTR with cooling coils and then adapting the effectiveness. The coolant energy could be expressed as follows

$$\dot{Q}_{cool} = \varepsilon_m \dot{m}_{cool} C_{p_{cool}} (T_{st} - T_{cool,in})$$ (6.64)

where Qcool is the energy flowrate removed through cooling system, mcool is the coolant mass flowrate, Cpcool is the heat capacity of the coolant, Tcool,in is the coolant inlet temperature and \(\varepsilon_m = 1 - e^{-NTU} = 1 - e^{-\frac{UA}{\dot{m}C_p}}\) Moreover it should be considered the cooling system as its temperature depends on the exchange heat. Besides the geometry of the cooling system plays a significant role for the evolution of the temperature and the exchanged heat. The temperature of the cooling liquid is warming along the fuel cell and be calculated by the following equation under the assumption of uniform temperature

$$\rho_{cool} V_{cool} C_{p_{cool}} \frac{dT_{cool}}{dt} = \dot{Q}_{cool} + \dot{m}_{cool} C_{p_{cool}} T_{cool,in} - \dot{m}_{cool} C_{p_{cool}} T_{cool}$$ (6.65)

where \(\rho_{cool}\) is the coolant density, Vcool is the coolant volume and Tcool is the outlet coolant temperature.

### 6.4.8. Air Compressor & DC Motor Model

The compressor presented in this section is a positive displacement, three-lobe roots type supercharger and is classified as a constant volume, variable pressure machine. This type of compressor is widely described in the literature, and a mathematical model is presented in Tekin (2006). Thus, the mass flow rate and rotation speed relation of the rotary vane compressor is stated as

$$\dot{m}_{comp} = \frac{1}{2\pi} \eta_{vol} \rho_{air} V_{comp}/rev \omega$$ (6.66)

where \(\eta_{vol}\) is the compressor’s volumetric efficiency, \(\rho_{air}\) is the air density and \(V_{comp}\) is the compressed volume per revolution. The power consumed by the screw compressor is then calculated for the inputs to the model using
the following expression for adiabatic compression

$$\dot{W}_{\text{comp}} = \tau_{\text{comp}} \omega_{\text{comp}} = c_{\text{air}} \dot{m}_{\text{comp}} \frac{T_{\text{in}}}{\eta_{\text{comp}}} \left[ \left( \frac{P_{\text{im}}}{P_{\text{atm}}} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right] \quad (6.67)$$

where $\tau_{\text{comp}}$ is the torque required to drive the compressor, $\omega_{\text{comp}}$ its rotational speed. It is assumed that there is no mass accumulation inside the compressor such that the mass flow into the compressor is equal to that leaving. The compressor torque can be expressed as

$$\tau_{\text{comp}} = \frac{\dot{m}_{\text{comp}} \omega_{\text{comp}}}{\eta_{\text{comp}}} \frac{T_{\text{atm}} c_{\text{air}}}{\eta_{\text{comp}}} \left[ \left( \frac{P_{\text{im}}}{P_{\text{atm}}} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right] \quad (6.68)$$

Using thermodynamic equations, the temperature of the gas stream leaving the compressor can be calculated from the temperature rise across the compressor.

$$T_{\text{out}} = T_{\text{atm}} + \frac{T_{\text{atm}}}{\eta_{\text{cp}}} \left[ \left( \frac{P_{\text{out}}}{P_{\text{atm}}} \right)^{\frac{\gamma - 1}{\gamma}} - 1 \right] \quad (6.69)$$

Furthermore, the change in humidity inside the compressor has been obtained using the following equation:

$$\phi_{\text{im}} = \frac{\phi_{\text{atm}} P_{\text{im}} P_{\text{sat}}(T_{\text{atm}})}{P_{\text{atm}} P_{\text{sat}}(T_{\text{im}})} \quad (6.70)$$

A lumped rotational parameter model with combined inertia is used to represent the dynamic behaviour of the compressor and electric motor speed. According to a Newton’s law, the polar second moment of area $J$ multiplied by the derivative of angular speed is equal the sum of all torques about the motor shaft.

$$J \frac{d\omega}{dt} = \sum \tau_i = \tau_{\text{motor}} - \tau_{\text{compressor}} \quad (6.71)$$

6.4.9. DC motor

The static model of the Lemco DC motor has been adopted from Grasser (2006). The motor power is calculated based on the voltage applied, which is the control input to the air supply system. The designed model could be used for various different motor technologies, and the applicable motor
equations are as follows

\[ V_{\text{applied}}(t) - V_{\text{emf}}(t) = L \frac{di}{dt} + Ri(t) \quad (6.72) \]

However, the back induced electromotive force (Vemf) is proportional to the angular speed (ω) seen at the shaft, \( V_{\text{emf}}(t) = K_b\omega(t) \), the previous equation becomes

\[ V_{\text{applied}}(t) = L \frac{di}{dt} + Ri(t) + K_b\omega(t) \quad (6.73) \]

where \( K_b \) is the emf constant that depends on the physical properties of the motor, \( i \) is the current, \( R \) is the nominal resistance(Ω), \( L \) is the nominal inductance(H) and \( J \) is the inertial load(Kg*m2/s2) Adopting state-space notation the combination of the above equations and using the speed and the current, the equations above are becoming

\[
\frac{d}{dt} \begin{bmatrix} i \\ \omega \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K_m}{J} & -\frac{K_f}{J} \end{bmatrix} \begin{bmatrix} i \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V_{\text{applied}}(t) \quad (6.74)
\]

where \( K_m \) is the armature constant, Finally, the torque motor (τmotor) is described by the following equation as it is proportional to the current induced by the applied voltage \( \tau_{\text{motor}}(t) = K_m i(t) \), \( K_m \) being the armature constant. Table 1 presents the values for the parameters of the mathematical model. The integrated system consists of 82 dynamic algebraic equations (DAE) and five degrees of freedom (DOF), which are the stack current, the mass flowrates of the coolant, anode and cathode channel, and the inlet temperature of the coolant. Figures 6.2 and 6.3 present the open loop simulation results of the PEM fuel cell system for the stack temperature and voltage during the start-up process and the FC steady state which is around 340K. Table 1. Constant coefficients used in the model
Figure 6.2.: Simulation for the stack and coolant temperatures

Figure 6.3.: Stack voltage
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>96500</td>
<td>C/mol</td>
</tr>
<tr>
<td>R</td>
<td>8.314</td>
<td>J/(mol.K)</td>
</tr>
<tr>
<td>$D_{H_2,ref}$ Hydrogen diffusion coefficient</td>
<td>9.15e-5</td>
<td>m²/s</td>
</tr>
<tr>
<td>$D_{O_2,ref}$ Oxygen diffusion coefficient</td>
<td>2.2e-5</td>
<td>m²/s</td>
</tr>
<tr>
<td>$D_{H_2O,ref}$ Water diffusion coefficient</td>
<td>34.5e-6</td>
<td>m²/s</td>
</tr>
<tr>
<td>$\Delta_r H^o(298K)$ Reaction enthalpy at 298K</td>
<td>-241820</td>
<td>J/mol</td>
</tr>
<tr>
<td>$\xi_1$</td>
<td>-0.9514</td>
<td></td>
</tr>
<tr>
<td>$\xi_2$</td>
<td>0.00312</td>
<td></td>
</tr>
<tr>
<td>$\xi_3$</td>
<td>-0.000187</td>
<td></td>
</tr>
<tr>
<td>$\xi_4$</td>
<td>7.4e-5</td>
<td></td>
</tr>
<tr>
<td>$\xi_5$</td>
<td>3.30e-3</td>
<td></td>
</tr>
<tr>
<td>$\xi_6$</td>
<td>-7.55e-6</td>
<td></td>
</tr>
<tr>
<td>$\xi_7$</td>
<td>1.10e-6</td>
<td></td>
</tr>
</tbody>
</table>

6.5. Model Identification

A model-based control strategy is a suitable approach to achieve optimal operation of the fuel cell and consequently to increase the efficiency of the system, but requires a capable dynamic fuel cell model. Indeed the crucial issue is to be able to balance the two counteracting conditions, namely ensuring acceptable response time for the power demand, while achieving high efficiencies over the entire operating range. Until now most dynamic two-phase PEM fuel cell models in the literature like Ziegler et al. (2005) are too complex for many process control purposes. Recently reduced order models have been derived, which are more suitable for model-based control (Golbert and Lewin (2004), Grötsch (2008), Del Real et al. (2007)). To design a controller suitable for our dynamic model, the first task will be to look into a reduced-order model suitable for model-based control. The next step will be to design a controller able to regulate the voltage output to a reference value, while the intensity is constant, and the system will be subject to different disturbances in the load. A reduced order SS model is designed with model identification from the data simulations of the PEM fuel cell process. The input/output data are obtained from simulation with
Simulink of the nonlinear system along the given set points, while the SS model parameters are obtained with Matlab Identification Toolbox by using parameter estimation method (PEM) for linear time invariant model, which is a standard prediction error/maximum likelihood method, based on iterative minimization of a criterion. The iterations are started up at parameter values that are computed from a subspace method that doesn’t use iteration. The parameterization of the system matrices follows a default canonical form. The mathematical representation of the SS model without disturbances and with sampling time equal to 1 sec is as follows

\[ x(t+1) = Ax(t) + Bu(t), y(t) = Cx(t) \]  \hspace{1cm} (6.75)

where \( y \) are the temperature and the voltage of the stack and \( u \) is the mass flowrate and the temperature of the coolant, and the voltage of the compressor. The system matrices are given as follows,

\[
A = \begin{bmatrix}
0.993 & 0.004 & 0.0009 & -0.001 \\
0.013 & 0.992 & 0.022 & 0.007 \\
-0.015 & -0.003 & 0.707 & -0.086 \\
-0.001 & -0.005 & 0.0069 & 0.986 \\
\end{bmatrix},
\]

\[
B = \begin{bmatrix}
-2.73e-06 & -0.00070165 & 2.090e-05 \\
-0.003405 & 0.0050739 & -2.53e-05 \\
0.0016905 & -0.042051 & -0.0001598 \\
-2.5e-05 & 0.0022222 & -4.27e-06 \\
\end{bmatrix},
\]  \hspace{1cm} (6.76)

\[
C = \begin{bmatrix}
394.3 & -8.715 & 0.0438 & -0.369 \\
45.10 & -19.16 & -0.112 & -0.104 \\
\end{bmatrix}
\]

A comparison between the reduced order SS model (green line) and the high-fidelity dynamic model (black line) is shown in Figures 6 and 7, for several operating conditions within the feasible stack temperature and voltage range. The SS model closely approximates the behaviour of the process.
model with a small approximation error (difference between the actual process output and the SS output) of 7\% (it is possible that this mismatch error will vary if the real fuel cell is used). In order to achieve output tracking, feedback controller is implemented with the assumption of disturbance mismatch between the system and the linear state space models. A more elegant way to overcome this error, would be to apply explicit/multi-parametric robust control methodologies such Pistikopoulos (2009).

Figure 6.4.: Different sequences of input variables

Figure 6.5.: Process and SS model comparison
6.6. PEM Fuel Cell Control Design

The next step in the general framework (Figure 2) involves the design of multi-parametric model predictive controller of the PEM fuel cell system. Model predictive control is based on online optimization of the future control moves, as opposed to other control methods which determine off-line a feedback policy. The survey paper of Mayne et al. (2000) provides a complete study of stability and optimality of constrained MPC. The main idea of MPC is to use a mathematical model of the system to predict the future effect of the control on the system behaviour (output horizon) and then compute the optimal sequence of manipulated variables (input horizon) that minimizes or maximizes the objective function and satisfies the constraints on inputs and outputs. In this work an explicit/multi-parametric MPC was developed (Pistikopoulos et al. 2007a Pistikopoulos and Dua (2007b), 2007b Pistikopoulos and Dua (2007a)) based on the reduced order linear model. In mp-MPC the optimization problem of the MPC is solved off-line by parametric optimization to obtain the optimal solution as an optimal mapping of the current state, output measurements and reference trajectory instead of demanding online optimization. The following MPC formulation is considered for the PEM fuel cell control system

\[
\min_{u_{t+1}, \ldots, u_{t+N_u}} J = \sum_{i=1}^{N_y} Q \left(y_i - y_{ref,i}\right)^2 + \sum_{j=0}^{N_u} R \left(u_j\right)^2
\]

s.t. \( x(t+1) = Ax(t) + Bu(t) \)

\[
y(t) = Cx(t) + d
\]

\[
\begin{bmatrix} 30 \\ 0.01 \\ 300 \end{bmatrix} \leq U_i \leq \begin{bmatrix} 150 \\ 0.8 \\ 340 \end{bmatrix} , \quad \begin{bmatrix} 25 \\ 300 \end{bmatrix} \leq Y_i \leq \begin{bmatrix} 60 \\ 360 \end{bmatrix}
\]

\( i = 1, \ldots, N_y, j = 0, \ldots, N_u - 1 \)

\( Q = 100, \quad \rho = 0.01, \quad R = 1 \)

(6.77)

where \( y \) is the controlled variables, \( u \) are the manipulated variables, \( N_u \) is the control horizon \((N_u = 2)\) and \( N_y \) the prediction horizon \((N_y = 10)\). The MPC takes into account the operational limitation of the manipulated variables (input constraints) and the controlled variables (output constraints). The problem involves six optimization variables.
\[m_{cool(t+1)}, T_{cool(t+1)}, V_{comp(t+1)}, m_{cool(t+2)}, T_{cool(t+2)}, V_{comp(t+2)}\]

and eight parameters \(\Theta = [x_t, T_{st}, V_{st}, T_{st,sp}, V_{st,sp}, T_{st,sp}]\)

which represent the states at time zero, the measurements of stack temperature and voltage and the reference trajectory of the stack temperature and the voltage. The objective function is set to minimize the quadratic norm of the error between the output variables and the reference points. The optimization problem (6.77) is a multi-parametric Quadratic Programming (mp-QP) problem and can be solved with standard multi-parametric programming techniques (Pistikopoulos et al., 2007a). In our study, the POP software was used (ParOS Ltd, 2003) to obtain the explicit controller description, which reacts in the optimal map of the control variables as function of the parameters of the system. This optimal map consists of 470 critical regions and the corresponding control laws. The projection of critical regions for the mp-MPC are shown in Figure 8. Each of the critical regions is described by a number of linear inequalities \(A_ix = b_i\) and its corresponding control is piecewise linear \(U_f = K_ix + c_i\), where \(i\) is the index of solutions. The critical regions are shown in Figure 6.6, based on a projection of the critical regions on the \(x_1 - x_2\) sub-spaces. Figures 6.7 and 6.8 depict the simulation results of the mp-MPC implementation for different operating conditions (set points). The controller manages to maintain the variables at the desired reference values for different values of current while satisfies the constraints. The controller showed fast response to temperature and voltage set point changes for different values of current while managed to keep the output constraints in the feasible range.

6.7. Concluding Remarks

In this chapter it is presented a dynamic mathematical model and a control design for the PEM fuel cell. This dynamic model provides results consistent with the literature, and well oriented towards control design. The first step in the proposed framework is to develop a high fidelity mathematical model for the PEM fuel cell system. Then a reduced order state space model is designed for optimal control studies. Finally a explicit/multi-parametric
Figure 6.6.: Projection on the state-space of the critical regions of mp-MPC

Figure 6.7.: Stack temperature
MPC controller has been developed to keep the controlled variables close to the set points while taking into account the physical constraints of the manipulated variables, namely the reactant and coolant mass flows. The designed controller shows good performance with respect to disturbance rejection.
7. Modelling and explicit model predictive control for PEM fuel cell

In this chapter we present the controller design for two different cases of PEM cell units. The first case involves the design of a constrained multi-parametric Model Predictive Control (mp-MPC) strategy for a fully automated integrated Hydrogen Fuel Cell Testing Unit (HFCTU) at CERTH. The process is described by a dynamic model which was developed, validated based on real temperature, pressure and voltage data on several operating conditions. This detailed mathematical model was used to derive reduced order state space models based on which explicit/multi-parametric MPC controller was designed in order to satisfy the load demand, avoid starvation and maintain the temperature at a nominal point. The derived controller was tested off-line on several operating conditions and showed an excellent transient and steady-state performance regardless the disturbances.

The second case, involves the temperature and oxygen excess ratio embedded controller of a PEM fuel cell system for 1.2 kW Ballard PEM fuel cell (Nexa Power Module) at University of Seville. The objective of this control system is to maintain the stack temperature and the oxygen excess ratio at a given set-point which is obtained from the results of a multi-objective optimization algorithm with the goal of minimizing the stack degradation and maximizing the net power. Due to the system delays, constraints and disturbances, two Explicit MPC is proposed to deal with this control problem. The simulation results show good performance of the controller which tracks the reference for the entire operating range.
7.1. Introduction

PEM (Polymeric Electrolyte Membrane) Fuel cells are devices which generate electric energy by combining hydrogen and oxygen through a polymeric membrane and produce water and heat. The low operating temperature and the fast start-up make this kind of fuel cells suitable for stationary and mobile applications. Currently this technology has been considerably developed but there are some drawbacks such as durability and production costs that need to be overcome in order to be competitive with conventional technologies. Thus, the application of advanced control techniques in order to improve efficiency and lifetime is justified.

The thermal dynamics plays an important role in the fuel cell performance. The work presented in Schmittinger and Vahidi (2008) shows that the operating temperature significantly influences on the water content inside the fuel cell. Therefore a low operating temperature causes a high water content which results in flooding of the anode channel Mckay et al. (2008) and reduces the fuel cell lifetime. Conversely, if the operating temperature is very high, the membrane becomes dehydrated which results in an increase in the rate of stack degradation. The authors conclude that there is an optimal range of water content that minimizes the degradation which corresponds to an optimal stack temperatures that will be the reference for the dynamic controller.

In the literature, the majority of previous contributions design and analyze air feed regulators which is as important as the temperature controller. Some authors (Riascos and Pereira (2009), Binrui et al. (2009), Ahn and Choe (2008) and Moré et al. (2010)) propose temperature controllers based on PID-controllers (Proportional Integral Derivative controller). Specifically, Riascos and Pereira (2009) presents a PI-controller with small gains, Binrui et al. (2009) develops a fuzzy incremental PID control algorithm, Ahn and Choe (2008) compares the results of a classic PI-controller with a state feedback controller and Moré et al. (2010) proposes a PI-controller with anti-windup. The work published in Na and Gou (2008) designs a feedback loop with disturbance compensator. In addition, the works presented in Riascos and Pereira (2009) and Binrui et al. (2009) include a generator of optimal temperature references based on the air humidity and load profile respectively. To the best of our knowledge, none of these papers designs an
advanced controller which is capable of dealing with physical constraints, disturbances and delays simultaneously in order to improve the control performance. To this end, this work focuses on controller design and performance and proposes an Explicit MPC (Model Predictive Control) controller Bemporad et al. (2002) which includes feedforward to take disturbances into account, predicts the temperature profile to overcome the delays and physical constraints and reduces the execution time making the solution of this control problem feasible for real-time implementation. Moreover, the set-points are obtained by a multi-objective optimization algorithm based on the water content which minimizes the degradation and maximizes the net power supplied. This optimization is not described here because it is not the scope of this work.

7.2. Multi-Parametric Model Predictive Control of an Automated Integrated Fuel Cell Testing Unit

7.2.1. Mathematical Modelling and Experimental Setup

The PEM fuel cell system is described by a dynamic mathematical model based on the work of Ziogou et al. (2010), which has been validated using experimental data in several operating conditions in order to guarantee the accuracy of the simulation results. The mathematical model used in this work is 0D and includes mass balances for the anode and cathode side and recirculation, semi-empirical equations for the membrane, electrochemical equations, heat balances for the fuel cell and mass and energy equation for the humidifier, compressor and the cooling system. The mathematical modelling of the mass transport phenomena along the channels, the gas diffusion layers and the membrane consist the five volume approach developed by Pukrushpan (2005). Moreover, a simple semi-empirical equation was used for the voltage calculation that accounts for temperature and the various voltage losses. The main structure of the model along with the mass transport flow is depicted in Figure 7.1.
7.2.2. Experimental System

The developed Fuel Cell Testing Unit (FCTU) of LPSDI/CERTH is comprised out of a humidification system, two mass flows for the regulation of the gases and two PID controllers for the anode and cathode pressure regulation. The temperature control subsystem includes a fan assisted air cooling system and an electrical heat up system. The automated operation and the data acquisition are conducted through an on-line supervisory control and data acquisition system (SCADA). The integrated system is equipped with an electronic load, which simulates the power demands or the power fluctuations that occur in real systems that use fuel cells for power generation.

The single PEM fuel cell uses a membrane electron assembly by ElectroChem (FC05-02SP) with a Nafion 1135 membrane and an active area
of 25cm². The experiments are conducted at 1Barg pressure and nominal
temperature of 338K.

7.2.3. Model Identification

Prior to the control framework (1.3) the non-linear model needs to be simpli-
ified in order to be used for the MPC controller design. Therefore a discrete
reduced order state space (SS) model for each control objective is obtained
using a model identification technique which maintains adequately the dy-
namic behaviour of the system. The input-output data are obtained from
simulations of the non-linear mathematical model for various operating and
the parameters of the SS models are determined from the Identification
Toolbox of Matlab (Figure 7.3). The sampling time for the data acquisition
is 100ms for the power, the current and the mass flow rates, and 1sec for
the temperatures, since is slower phenomenon. The mathematical represen-
tation of the SS model with additive disturbances is as follows

\[ \dot{x}_{t+1} = Ax_t + Bu_t + Cv_t, \quad y_t = Dx_t \quad (7.1) \]

The first SS model approximates the behaviour of the power and has 1 ma-
 nipulated variable (I), one control variable (P) and one known disturbance,
the difference between the maximum power and the measured power (∆P)
and two states.

\[ A_1 = \begin{bmatrix} 0.99912 & 0.028574 \\ 0.0054676 & 0.82201 \end{bmatrix}, B_1 = \begin{bmatrix} 0.0059061 \\ -0.036945 \end{bmatrix} \]

\[ C_1 = \begin{bmatrix} -0.00018942 \\ 0.0016459 \end{bmatrix}, D_1 = [12.565 \quad -0.0015507] \]

The second and the third reduced order models are approximate the be-
 haviour of the oxygen and hydrogen excess ratio. Both SS model have 1
state, 1 disturbance (I), one manipulated variable which is the excess of
oxygen (\(\lambda_{O2}\)) and hydrogen (\(\lambda_{H2}\)) and one output, the mass flow rate of air
(\(\dot{m}_{air}\)) and hydrogen (\(\dot{m}_{H2}\)), respectively. The system matrices are given as
follows

\[ A_2 = [-0.014276], B_2 = [0.060283], C_2 = [-0.0088574], D_2 = [233.68] \]
A_3 = [-0.055887], B_3 = [0.0086835], C_3 = [-0.0055123], D_3 = [238.05]

The fourth SS model represents the behaviour of the temperature which is maintained at the desired set point through a heatup and a cooling system. The system matrices are: presented below

A_4 = 1, B_4 = [0.0052417 - 0.0004687], C_4 = [3.3348e - 009], D_4 = 1

This model has one state, one output (T), one known disturbance the ambient temperature (T_{amb}) and two inputs, the power to the resistance for the heat-up (W_R) and the power of the fans (W_{cl}), which both are translated to percentage of the full equipment operation. We have to note here that we design 4 state space models, based on which we designed the controllers, because the attempt to developed one accurate was not possible due to identification error. It might be possible to develop only one if we used different model reduction technique.

7.2.4. Multi-Parametric Model Predictive Controller (mp-MPC)

The next step involves the design of multi-parametric Model Predictive Controllers of the underlying PEMFC system. In model predictive control (MPC), an open-loop optimal control problem is solved at regular intervals, given the current process measurements, to obtain a sequence of the current and future control actions up to a certain time horizon, based on the future predictions of the outputs and/or states obtained by using a mathematical representation of the controlled system. The following optimization problem is considered for the design of the PEM fuel cell control system,

\[
\begin{align*}
\min_{x,u,y} J &= \sum_{i=1}^{N_y} (y_i - y_{sp,i})^T Q (y_i - y_{sp,i}) + \sum_{j=0}^{N_u-1} (u_j)^T R (u_j) \\
\text{s.t.} \quad &x(t + 1) = Ax(t) + Bu(t) + Cv(t) \\
&y(t) = Dx(t) \\
&y_{\min} \leq y(t) \leq y_{\max}, u_{\min} \leq u(t) \leq u_{\max}, v_{\min} \leq v(t) \leq v_{\max}
\end{align*}
\]

(7.2)

where \( u \) are the manipulated variables, \( y \) are the controlled variables, \( Nu \) is the control horizon and \( Ny \) the prediction horizon. The objective function is set to minimize the quadratic norm of the error between the output variables
and the reference points. Moreover, the system includes physical constraints which should be satisfied during the operation:

- $0.1 \text{A} \leq I \leq 12 \text{A}$, $500 \text{cc/min} \leq m_{\text{air}} \leq 3000 \text{cc/min}$
- $0 \leq W_{R} \leq 55.8 \text{W}$, $313 \text{K} \leq T_{\text{fc}} \leq 353 \text{K}$
- $0.1 \text{W} \leq P \leq 7 \text{W}$, $200 \text{cc/min} \leq m_{H_{2}} \leq 1000 \text{cc/min}$
- $0 \leq W_{d} \leq 25.8 \text{W}$, $288 \text{K} \leq T_{\text{amb}} \leq 313 \text{K}$

The aforementioned optimization problem (7.2) is a multi-parametric Quadratic Programming problem and can be solved with standard multi-parametric techniques (mp-QP). In our study, the explicit parametric controller was derived with the Parametric Optimization (POP) software. The control horizon in each problem is 2, therefore there are two optimization variables ($u_{t+0}, u_{t+1}$).

The corresponding parameters of each problem are shown in Table 7.1 with the respective number of explicit/multi-parametric MPC controller’s critical regions, while Figure 7.3 presents the control design for the PEMFC system including the input/output variables of each controller and the interactions between them.

### Table 7.1: Optimization problem parameters and settings

<table>
<thead>
<tr>
<th>Output variable (y)</th>
<th>Optimization variables $u$</th>
<th>Parameters($\theta$)</th>
<th>Horizon($N_{y}$)</th>
<th>$CR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$I(t+0), I(t+1)$</td>
<td>$[x_{1}, x_{2}, I, \Delta P, P_{sp}]$</td>
<td>10</td>
<td>67</td>
</tr>
<tr>
<td>$\lambda_{O_{2}}$</td>
<td>$m_{O_{2}}(t+0), m_{O_{2}}(t+1)$</td>
<td>$[x_{1}, I, \lambda_{O_{2}}, \lambda_{O_{2},sp}]$</td>
<td>20</td>
<td>13</td>
</tr>
<tr>
<td>$\lambda_{H_{2}}$</td>
<td>$m_{H_{2}}(t+0), m_{H_{2}}(t+1)$</td>
<td>$[x_{1}, I, \lambda_{H_{2}}, \lambda_{H_{2},sp}]$</td>
<td>40</td>
<td>13</td>
</tr>
<tr>
<td>$T_{\text{fc}}$</td>
<td>$W_{R}(t+0), W_{R}(t+1)$</td>
<td>$[x_{1}, T_{\text{amb}}, T_{\text{fc}}]$, $T_{\text{fc},sp}$</td>
<td>100</td>
<td>17</td>
</tr>
</tbody>
</table>

7.2.5. Simulation Results

The simulation results of the mp-MPC implementation for several operating conditions are shown in Figures 7.4, 7.5 and 7.6. During the simulation, we assumed that the ambient temperature was constant at the value of 298K. The performance of the temperature controller is presented in Figure 7.4, for three temperature set point changes ($333 \text{K}, 338 \text{K}, 343 \text{K}$), while the power demand remains constant (5W), and it is observed that the controller fol-
Figure 7.3.: Control structure

Figure 7.4.: Temperature control and cooling/heatup
lows rapidly the set point changes on the temperature without offset. The PEM Fuel Cell system needs to be heated during steady state operation in order to follow the set point. In Figures 7.5 and 7.6 the performance of the power and oxygen excess ratio controllers is presented. During the experiment the hydrogen excess ratio is constant ($\lambda_{H2}=1.5$) and the temperature at 338K. The mass flow rates are properly adjusted to fulfill the starvation avoidance constraint by keeping the excess ratio at constant level. The power controller showed excellent response to load changes and the excess ratio controller demonstrated fast settling time (less than 2s) after current disturbances.

![Figure 7.5: Control of power and $\lambda$ O2](image)

The four explicit/multi-parametric MPC controllers are able to track the desired reference points regardless the fluctuations of the interacting variables. The controllers manage to fulfill the power demand, while avoiding starvation and maintain the fuel temperature at the desired set point. Finally, the system response is within the feasible area of operation since the output of the controllers was bounded by the operating constraints and the stability was guaranteed.
7.3. Design of a MPC controller for regulating temperature in PEM fuel cell systems considering stack degradation

This section studies the influence of stack temperature and oxygen excess ratio on the stack performance and net power generation and proposes an advanced controller which maximizes the fuel cell lifetime and efficiency. As previously mentioned, normal operation of the fuel cell system relies on an exothermic reaction, which has the effect of increasing the temperature of the fuel cell. As the stack temperature rises, so does the rate of reaction, which in turn results in an increase in net power generation and the overall efficiency of the fuel cell. However, excessive increases of fuel cell temperature result in the degradation of the membrane and the fuel cell stack due to thermal stress and degradation. Ultimately, a compromise between degradation and efficiency must be obtained in order to appropriately control the operation of the fuel cell.

7.3.1. System Description

The system under consideration in this section is based on a 1.2 kW Ballard PEM fuel cell (Nexa Power Module). This apparatus is currently in use by many research groups and considered to be the state of the art in PEM
technology. The stack of this PEM fuel cell comprises 46 cells, each with a 110 cm$^2$ membrane. The system is auto-humidified and air-cooled by a small fan which consumes as a maximum over 100 W. Hydrogen was fed to the fuel cell in dead-end mode, with flushing. Figure 7.7 shows the real fuel cell system which is installed in the laboratory of the University of Seville.

Figure 7.7: 1.2 kW Nexa power module

### 7.3.2. Non-linear PEM fuel cell model

A semi-empirical mathematical model was developed in previous work Del Real et al. (2007) which includes two-phase fluid dynamics considering flooding phenomenon, heat transfer dynamics and algebraic formulation of the polarization curve.

The thermal equations of the PEM fuel cell model are summarized here, which has been validated through experimental data of the bed-test. An energy balance is performed in order to obtain the thermal model, accounting for the energy rate produced in the chemical reaction of water formation (which is supposed to be formed as water vapor), $\dot{H}_{reac}$, the power supplied in the form of electricity, $P_{st}$, and the amount of heat evacuated by radiation, $\dot{Q}_{rad,B2amb}$, and both natural and forced convection, $\dot{Q}_{conv,B2amb}$. Heat removal is completed through forced convection by a small fan. In bigger fuel cell stack systems, where the amount of heat is considerably larger, water cooling is necessary. In those cases, the forced convection term should be substituted by other terms which model heat exchange in cooling fluid.
The energy balance results in:

\[ m_{st} \cdot C_{st} \cdot \frac{dT_{st}}{dt} = \dot{H}_{\text{reac}} - P_{st} - \dot{Q}_{\text{rad},B2amb} - \dot{Q}_{\text{conv},B2amb} - \dot{Q}_{\text{conv},B2amb} \cdot . \quad (7.3) \]

The enthalpy flow rate, where \( h^0_{f,H_2O(g)} \) is the mass specific enthalpy of formation of water vapor and \( c_p,H \), \( c_p,O_2 \), and \( c_p,H_2O(g) \) are the specific heats of hydrogen, oxygen and water vapor respectively, will be:

\[ \dot{H}_{\text{reac}} = m_{H_2,\text{anGDL}} \cdot \Delta h_{H_2} + m_{O_2,\text{caGDL}} \cdot \Delta h_{O_2} - W_{H_2O,\text{gen}(g)} \cdot (h^0_{f,H_2O(g)} + \Delta h_{H_2O(g)}) \cdot , \] (7.4)

\[ \Delta h_{H_2} = c_p,H_2 \cdot (T_{\text{anch,in}} - T^0) \cdot , \] (7.5)

\[ \Delta h_{O_2} = c_p,O_2 \cdot (T_{\text{cach,in}} - T^0) \cdot , \] (7.6)

\[ \Delta h_{H_2O(g)} = c_p,H_2O(g) \cdot (T_{st} - T^0) \cdot , \] (7.7)

where \( T^0 \) is the reference temperature for the enthalpy, \( T_{\text{anch,in}} \) is the inlet temperature of the anode channel, \( T_{\text{cach,in}} \) is the inlet temperature of the cathode channel and \( T_{st} \) is the stack temperature.

Energy produced in the form of electricity is given by:

\[ P_{st} = V_{st} \cdot I_{st} \cdot , \quad (7.8) \]

where \( V_{st} \) is the stack voltage, \( I_{st} \) is the stack current.

Heat exchanged as radiation is modeled as shown below:

\[ \dot{Q}_{\text{rad},B2amb} = \varepsilon \cdot \sigma \cdot A_{B2amb,rad} \cdot (T_{st}^4 - T_{amb}^4) \cdot , \quad (7.9) \]

where \( \varepsilon \) is the emissivity, \( \sigma = 5.678 \cdot 10^{-8}[Wm^{-2}K^{-4}] \), the Stefan–Boltmann constant, \( T_{amb} \) is the ambient temperature and \( A_{B2amb,rad} \) is the radiation exchange area.

Lastly, the convective heat flux is given by the sum of the natural convection, \( \dot{Q}_{\text{conv},B2amb,nat} \), and the forced convection, \( \dot{Q}_{\text{conv},B2amb,forc} \). In each case, the convective heat transfer coefficients (\( h_{B2amb,nat} \) and \( h_{B2amb, forc} \)) are different, similarly to the exchange areas (\( A_{B2amb,conv} \) and \( A_{B2cool,conv} \)), as natural convection takes place in the fuel cell lateral walls, and forced
convection occurs across the internal walls of the cells, which are constructed as a radiator.

\[ Q_{\text{conv,B2amb}} = Q_{\text{conv,B2amb,nat}} + Q_{\text{conv,B2amb,forc}} , \]  
\[ (7.10) \]

\[ Q_{\text{conv,B2amb,nat}} = h_{B2amb,nat} \cdot A_{B2amb,\text{conv}} \cdot (T_{st} - T_{amb}) , \]  
\[ (7.11) \]

\[ Q_{\text{conv,B2amb,forc}} = h_{B2amb,\text{forc}} \cdot A_{B2cool,\text{conv}} \cdot (T_{st} - T_{cool}) , \]  
\[ (7.12) \]

where

\[ h_{B2amb,\text{forc}} = K_{h1} \cdot (\dot{m}_{cool})^{K_{h2}}. \]  
\[ (7.13) \]

\( T_{cool} \) is the temperature of the cooling air and \( \dot{m}_{cool} \) is the cooling air flow.

Regarding air cooling, a small fan is used to supply the cooling air flow. The heat transfer dynamics of the fuel cell are several magnitude orders slower than the fluid-dynamics associated with the cooling air flow, therefore, the last ones are neglected in our model. Moreover, the amount of air supplied by the fan can be considered as linearly proportional to the control signal of the fan. In this way, the equation that links the fan voltage, \( V_{fan} \), between 0 and 100 (%), with the air flow supplied expressed in kg s\(^{-1} \), \( \dot{m}_{cool} \), is given by:

\[ \dot{m}_{cool} = 36 \cdot V_{fan} . \]  
\[ (7.14) \]

The dynamical mathematical model of the PEM fuel cell system is implemented in MATLAB/Simulink and is validated with experimental data gathered from the Nexa Power module. Figure 7.8 shows the comparison of the heat transfer dynamics of the model and of the bed-test. Note that the dynamic of the thermistor is not included in the model causing a slight difference between both curves.

The thermistor is a type of resistor whose resistance varies with temperature and it is used as temperature sensor. This sensor can be modeled by a first-order approximation with a characteristic time of 10 seconds.

\[ \tau \cdot \frac{dT_{st}^*}{dt} + T_{st}^* = T_{st} , \]  
\[ (7.15) \]

where \( T_{st}^* \) is the temperature measured by the thermistor and \( \tau \) is the characteristic time of the thermistor.
7.3.3. Discrete Linear model

Two discrete time state space (SS) models are obtained for the MPC controller design using system identification techniques on the non-linear model presented in Section 7.3.2. The first (SS) model is designed to approximate the behaviour of the stack temperature, and has as manipulated variables (inputs) the fan voltage, $V_{fan}$, and as output variable the stack temperature, $T_{st}$. Moreover, this model has two known disturbances which are the load current, $I_{st}$, and the ambient temperature, $T_{amb}$ which is equal to the temperature of the cooling air. The second linear SS model is designed to approximate the behaviour of the oxygen excess ratio, $\lambda_{O_2}$, by using as manipulated variable the voltage of the compressor, $V_{comp}$, and as known disturbance the stack current (Figure 7.10).
Simulations were performed on the non-linear model with Simulink and the input/output data were obtained with sampling time $8ms$. The results of the simulation are shown in Figures 7.9 and 7.10. This data were used to perform system identification and obtain the two SS models. Both SS models consist of two state variables ($x$), one input ($u$) and one output ($y$). A comparison between the reduced order SS models and the high-fidelity dynamic model is shown in Figures 7.9 and 7.10. The SS models closely approximates the behavior of the process with a small approximation error (difference between the actual process output and the SS output) of 7% and 6% respectively.
The mathematical description of the two linear SS models is given below

\[ x(t+1) = A \cdot x(t) + B \cdot u(t) + C \cdot v(t) \]
\[ y(t) = D \cdot x(t) \]

where the system matrices for the stack temperature model are

\[
A_1 = \begin{bmatrix} 0.99 & -0.006 \\ -0.002 & 0.98 \end{bmatrix}, B_1 = \begin{bmatrix} -0.0000008 \\ -0.000014 \end{bmatrix}, \\
C_1 = \begin{bmatrix} -0.0000048 & -0.000005 \\ 0.0000007 & -0.000001 \end{bmatrix}, D_1 = \begin{bmatrix} 319.2 & -157.3 \end{bmatrix}
\]

The system output \( y \) is the stack temperature, \( T_{st} \), the input \( u \) is the fan voltage, \( V_{fan} \), the known disturbance vector is equal to \( [I_{st}, T_{amb}] \) and the sampling time is 0.008 sec.

The system matrices for the oxygen excess ratio model are

\[
A_2 = \begin{bmatrix} 0.93 & 0.26 \\ 0.26 & -0.11 \end{bmatrix}, B_2 = \begin{bmatrix} 0.004 \\ -0.016 \end{bmatrix}, \\
C_2 = \begin{bmatrix} -0.013 \\ 0.05 \end{bmatrix}, D_2 = \begin{bmatrix} 22.06 & -0.166 \end{bmatrix}
\]
The system output \(y\) is the oxygen excess ration, \(\lambda_{O_2}\), the input \(u\) is the compressor voltage, \(V_{\text{comp}}\), the known disturbance is the stack current, \(I_{\text{st}}\), and the sampling time is 0.008 sec.

### 7.3.4. Explicit MPC formulation

This section involves the design of explicit/multi-parametric Model Predictive Controllers (mp-MPC) of the PEM fuel cell system. A nominal mp-MPC controller is designed Bemporad et al. (2002), based on the linear state-space model obtained in the previous section by considering \(A_d\), \(B_d\), \(D_d\) and \(C_d\) as constant matrices. The following MPC formulation is considered for the PEM fuel cell system:

\[
\begin{align*}
\min_{x,u,y} & \quad \sum_{k=1}^{N_y-1} \left[ (y_k - y_{\text{ref},k})^T \cdot Q \cdot (y_k - y_{\text{ref},k}) \right] + \\
& \quad + \sum_{k=1}^{N_u-1} \left[ (u_k - u_{\text{ref},k})^T \cdot R \cdot (u_k - u_{\text{ref},k}) \right] + \\
& \quad + (y_{N_y} - y_{\text{ref},N_y})^T \cdot P \cdot (y_{N_y} - y_{\text{ref},N_y}),
\end{align*}
\]

subject to

\[
\begin{align*}
\mathbf{x}(k+1) &= A_d \cdot \mathbf{x}(k) + B_d \cdot u(k) + D_d \cdot \mathbf{v}(k), \\
y &= C_d \cdot \mathbf{x}, \\
y_{\text{min}} &\leq y(k) \leq y_{\text{max}}, \\
u_{\text{min}} &\leq u(k) \leq u_{\text{max}}, \\
v_{\text{min}} &\leq \mathbf{v}(k) \leq \mathbf{v}_{\text{max}},
\end{align*}
\]

where \(u\) is the manipulated variable, \(y\) is the controlled variable, \(y_{\text{ref}}\) is the optimal temperature profile, \(u_{\text{ref}}\) is the input reference profile, \(N_y\) is the prediction horizon and \(N_u\) the control horizon (\(N_y = 10, N_u = 2\)).

The first optimization problem involves two optimization variables \(ut+0, ut+1\) and eight parameters \(\mathbf{x} = [x_1, x_2, ut-1, v_1, v_2, y_{\text{real}}, y_{\text{ref}}, u_{\text{ref}}]\) which represent the states at time zero, the previous value of the fan voltage, the two known disturbances(\(T_{\text{amb}}, I_{\text{st}}\)), the actual temperature value and the reference profiles for temperature and fan. The second optimization problem involves two optimization variables \(ut+0, ut+1\) and eight parameters
x = [x_1, x_2, u_{t-1}, v_1, y_{real}, y_{ref}] which represent the states at time zero, the previous value of the compressor voltage, the known disturbance (I_{st}), the actual value and the reference profiles of the oxygen excess ratio. The objective function is set to minimize the quadratic norm of the error between the output and its optimal profile while the constraints on u, v and y are also introduced. Particularizing the physical constraints are:

\[\begin{align*}
300 \, K &\leq T_{st} \leq 333 \, K, \\
0 &\leq V_{fan} \leq 100, \\
0 &\leq I_{st} \leq 40 \, A, \\
290 \, K &\leq T_{amb} \leq 305 \, K.
\end{align*}\]

For the case of constant system matrices, the optimization problem is a multi-parametric Quadratic Programming (mp-QP) problem and can be solved with standard multi-parametric programming techniques Pistikopoulos and Morari (2002a). In our study, Parametric Optimization Software was used PAROS (2009) to obtain the explicit controller description, which is the optimal map of the control variables as function of the parameters of the system. This optimal map consists of 7 critical regions and the corresponding control laws for the temperature controller. Each of the critical regions is described by a number of linear inequalities \( A_i \cdot x \leq b_i \) and its corresponding control action is piecewise linear \( u = K_i \cdot x + c_i \), where \( i \) is the index of solutions. The critical regions for the temperature controller are shown in (Figure 7.11), based on a projection of the critical regions on the \( T_{ref} − T_{real} \) sub-spaces and for the oxygen excess ratio are shown in (Figure 7.12) based on the projection on the \( \lambda_{O_2} − \lambda_{O_2, ref} \).
Figure 7.11.: Critical Regions for Stack temperature set point - Stack temperature
Figure 7.12.: Critical Regions for Oxygen excess ratio set point- Oxygen excess ratio

Figure 7.13 shows the temperature control scheme which comprises four blocks: temperature reference generator, water content and temperature controller. Note that the water content observer and the temperature reference generator is included in the scheme, the water content, $m_l$, is estimated by an observer which is a function of the voltage drop and the load current while the temperature reference generator calculates the reference temperature in order to maintain the water content in a range which minimizes the membrane degradation and to simultaneously maximize the stack net power.
7.3.5. Simulation results

In this section, the simulation results are presented and discussed. The fuel cell is simulated including the oxygen excess ratio controller which was published in Arce et al. (2009). This controller has been implemented in real-time and validated with experimental data. The control objective is to track the oxygen excess ratio, $\lambda_{O_2}$, which is a variable related to the starvation phenomenon. The oxygen excess ratio is estimated by an observer which is a function of the stack temperature measured, $T_{st}^*$, the air flow which enters into the cathode, $\dot{m}_{ca,in}$ and the load current, $I_{st}$. Moreover, the oxygen excess ratio reference, $\lambda_{O_2,ref}$, is calculated by a reference generator which interpolates a curve obtained off-line by an optimization algorithm. The optimal oxygen excess ratio depends on the load current and stack temperature. Furthermore, the controller is an explicit MPC controller which includes disturbances and physical constraints, and the execution time is short enough to make this proposal feasible for real-time implementation as it was demonstrated in Arce et al. (2009). Regarding the remaining actuators, the hydrogen valve is regulated by a proportional controller whose objective is to keep constant the anode pressure and the purge valve control is a heuristic controller which opens the valve when the voltage drop measured corresponds to a certain amount of water. Figure 7.17 and 7.18 depict the simulation results of the controller for different set point changes of the
oxygen excess ratio. The controller manages to maintain the variables at the desired reference values for different values of current while satisfies the constraints.

Figure 7.14 shows the global control scheme where the oxygen excess ratio control loop and the temperature control loop are represented in detail. The disturbances of each controller are clearly shown. As may be seen, the input of the controlled fuel cell system is the current demanded by the electronic load and the output is the net stack power supplied by the fuel cell. Due to the fact that the simulation results are focused on the performance of the temperature controller, the temperature optimization is not included in the simulations which will be studied in future works.

![Control of the fuel cell system scheme](image)

Figure 7.14.: Control of the fuel cell system scheme

Figure 7.15 and 7.16 depict the simulation results of the mp-MPC implementation for different operating conditions (set points). The controller showed fast response to temperature set point changes for different values of current and ambient temperature while managed to keep the hard operational output constraints in the feasible range.

**Load current effects**

The effect of the load current variation is analyzed. Figure 7.15 shows the load current and the ambient temperature profiles simulated. The ambient temperature changes between 293K and 300K, and the load current is varied from 20A to 30A. In Figure 7.18, the simulation results of the fan voltage
and oxygen excess ratio are presented.

The performance of the stack temperature controller is presented in Figure 7.15 and Figure 7.16, and for the oxygen excess ratio controller is presented in Figure 7.17 and Figure 7.18, where the stack temperature, the fan voltage, ambient temperature and load current are presented. The controller suitably tracks the outputs at the desired values under different values of disturbances, while satisfies the constraints.

The effects of the load current is observed on the fan voltage action which increases for higher currents and decreases for lower currents because the increment of current demand generates more heat that has to be removed. Moreover, the increase of the current has a direct effect on the compressor voltage since the oxygen consumption is increased.
Ambient temperature effects

The air which is used to cool the stack down by the fan is at the same temperature of the ambient. Thus, the ambient temperature influences the heat transfer dynamics. The fuel cell stack needs more air in order to maintain the same stack temperature, if the ambient temperature is higher. In Figure 7.15 and 7.16, the effects of the ambient temperature on the control action is observed. The ambient temperature is varied from 293K and 300K. As can be seen the reference is perfectly tracked and there is no steady-state error despite the variations at the ambient temperature. Furthermore, as it can be seen from Figure 7.17 and Figure 7.18 the ambient temperature doesn’t have any effect on the oxygen excess ratio and consequently on the compressor voltage, thus our assumption on the design of the oxygen ratio controller was correct.
7.3.6. Concluding Remarks

This work presents the design of two Explicit MPC controllers in order to regulate the stack temperature and the oxygen excess ratio of a PEM fuel cell system. The reduced order models designed by system identification techniques and closely approximate the behaviour of the non-linear model of the PEM fuel cell, with error 6% and 7%, respectively. This approximation error could be introduced as an uncertainty in the linear model, which could be taken into account during the (robust) controller design in a future work. Both explicit/multi-parametric MPC controllers suitably track the outputs,
oxygen excess ratio and stack temperature, at the desired values for several set points and under different values of disturbances, while satisfies the hard operational constraints.
8. Experimental Unit of PEM Fuel Cell System

In this chapter we present the analytical design of the PEM Fuel Cell Unit for 1kW PEM fuel cell stack from NEDSTACK, which includes the Process Instrumentation Diagram, the list of the materials and the specification of the designed unit. Moreover, based on the experimental results provided by the manufacturer, a validated dynamic mathematical model of the PEM fuel cell unit is presented.

8.1. Introduction

PEM Fuel cell transforms the the chemical energy of the hydrogen into electrical energy with high efficiency and low environmental impact, while produces water and heat as by-products. The basic physical structure of a fuel cell consists of an electrolyte layer in contact with anode and cathode on both sides. The fuel cell is fed continuously to the anode (negative electrode) and an oxidant (often oxygen from air) is fed continuously to the cathode (positive electrode). The electrochemical reactions take place at the electrodes to produce an electric current through the electrolyte, while driving a complementary electric current that perform work on the load. The unit is designed for use with normal temperature PEM fuel cell stack by NedStack (1kW power). The power demand will be simulated by Electronic load of EleKtroAutomatik (EL9000 series) with serial communication enabled. The overall system will be monitored by a PC-based control platform by National Instruments (PXI series). Finally, water cooling system will be used to heat-up the fuel cell during the start-up purposes and cool-down during the operation. A simple schematic representation of a unit cell is shown in Figure 8.1, while Figure 8.2 presents the analytical design which includes all the sensors, flowmeters, units and transmitters. The full list of
materials as well as three dimension (3D) figures of the units are presented in Appendix II.

8.2. Equipment

8.2.1. PEM Fuel Cell

The fuel cell stack is integrated with several auxiliary components to form a complete fuel cell system (8.1,8.2). The fuel cell stack requires four flow system: i) hydrogen supply to the anode, ii) air supply to the cathode, iii) de-ionized water serving as coolant in the cooling channel and iv) de-ionized water to humidify the hydrogen and the air flow. The reactant flow, the stack temperature and pressure, the membrane humidity are critical parameters which require a precise control to the viability, efficiency and robustness of the fuel cell propulsion systems. The table 8.2.1 presents the main specifications of the 1kW NEDSTACK PEM fuel cell, while the full list of the specifications for the operating temperature, fuel and air flow rates is presented in Appendix II.
Figure 8.2.: Detailed Process Instrumentation Diagram

168
8.2.2. DC LOAD

The power management subsystem controls the power drawn from the fuel cell stack. The load current can be seen as a disturbance to the fuel cell system to be controlled. The DC electronic load (EleKtroAutomatik EL9000) is used for testing purposes of the PEM fuel cell behaviour. Research and development in the field of fuel cells, demands electronic loads that allow a current to flow even at 0V, as well as load systems with expandable capacity for testing stacks of cells. This type of electronic load offers the option to remotely controlled via an interface card and monitored from a PC. The electronic load can be operated in conjunction with a power supply or with the additional analogue interface socket at the rear side, controlled and monitored by an other device with an analogue interface. Electronic loads can have programmable characteristics depending on the output load required.

1. Constant Current Mode: In this mode the load current is constant even though the voltage at the terminals of the load changes

2. Constant Voltage Mode: In this mode the load voltage is constant even though the current into the load changes

3. Constant Resistance Mode: The load behaves like a constant resistance
4. Constant Power Mode: The load power is kept constant

8.2.3. Transmitters

Humidity and Temperature transmitters

The humidity and temperature transmitters are designed for monitoring relative humidity and temperature in building energy management systems. The combination of high accuracy, stability and reliable operation, make these products the ideal choice for demanding applications. The humidity and temperature transmitters are two-wire transmitters. They are duct mounted, and the electronics can be disconnected without dismantling the installation.

Pressure transmitters

The pressure transmitters (or transducers) are of the strain gauge type in a Wheatstone bridge configuration. The output of the sensor is a function of both the supply voltage to the bridge and the input which causes an imbalance in the bridge. An excitation voltage of 24V is provided and this results in a full-scale output of 4-20 mA (or 0-5 VDC). Accuracy, including non-linearity, repeatability and hysteresis, is typically Â±0.25. They are also characterized by an offset current (or voltage), which is the output at zero input.

Level transmitters

Level transmitters fall into the same category as pressure transmitters. Usually, the bridge resistance is adjusted so that the output signal is zero millivolts with no weight applied. The voltage level of the output signal will then be directly proportional to the weight.

8.2.4. Controllers

Thermal Mass Flow Controller

In the case of mass flow control, the output signal is continuously compared with a setpoint signal from a voltage source. Any deviations between set-point signal and measured signal are translated into a solenoid control
valve adjustment until the two signals are identical. This type of mass flow controlling device consists of two separate items. The first component is the measuring device which consists of a parallel placed capillary, through which a part of the flow occurs, and the main flow channel. The output is generated by sensing the temperature gradient in the capillary tube before and after the gas is slightly heated up. The second component is a control valve which controls the flow through this device. A built-in controller controls the required flow. Typical output of this device is 5 VDC full-scale.

**Flow Meters**

The liquid sealed, rotating drum wet meter is constructed of a sealed hollow case in which a hollow drum of several compartments is free to rotate. The passage of gas through the meter causes the drum to rotate. The drum is designed so that when the case is filled with liquid to a predetermined level, each compartment will be filled and emptied. The meter is magnetically supported with an encoder, which send 1000 impulses per revolution of the drum to the computer.

**8.2.5. Valves**

**Control Valves**

Control valves have a continuant operation between fully open and fully closed. The fully open state of the control valve gives resistance to process line flow and cannot be expected to perform like a shut-off valve in the fully open state. The normal control range for electrical operated control valves is 4-20 mA. The built-in I/P converter converts this electrical signal in an equal air signal. This air signal acts on the diaphragm, which is directly connected to valve trim and allows to open and close the valve orifice. The amount of flow through the valve depends directly on the pressure. This pressure exerts a force on one side of the diaphragm against a spring which is located on the opposite side of the diaphragm. In an air to open control valve, the force exerted by the spring on the diaphragm is directed to close the valve and the control pressure force is exerted on the diaphragm to open the valve. An air to close air-operated control valve spring exerts a force on the diaphragm to open the valve, while the control pressure force is exerted on the opposite side of the diaphragm to close the valve.
Solenoid Valves

The solenoid valve is an electromagnet with a sliding magnetic core (plunger). When the current is applied to the coil, the plunger either opens or seals the orifice. When the coil is de-energized, the plunger returns to its original position. In a normally closed valve, the spring holds the plunger against the orifice. The spring pushes the plunger away from the orifice in a normally open valve. The electromagnetic force must not only overcome the force of the spring, but also any system fluid working against it. The magnitude of the electromagnetic force is a function of the diameter of the wire in the solenoid coil, the amount of current flowing through the coil and the number of turns in the coil.

8.2.6. Pumps

Piston/Diaphragm Pumps

Piston pumps are equipped with suction and discharge check valves. The pump is equipped with a stroke control which can be adjusted from the computer to increase or decrease the flow rate. The pumps are calibrated at several discharge pressures to achieve the required level of accuracy. Diaphragm pumps are basically identical except for the existence of an isolating diaphragm to avoid contact of the process fluid with the plunger. The diaphragm is driven by a hydraulic fluid which is pressurized by a piston.

Syringe Pumps

Syringe pumps consist of a syringe, a drive carriage and a gear drive. By moving the drive carriage at a constant rate with the gear drive, the plunger of the syringe is forced into the constant inner diameter of the syringe at a constant rate, producing a constant fluid flow rate. Syringe pumps are able to achieve very low and very constant flow rates. The period of time that the pump can continuously run is limited by the capacity of the syringe and the magnitude of the volumetric flow rate.
8.2.7. Other Equipments

Thermocouples

Thermocouples are bimetallic junctions which generate a thermal electromotive force proportional to the junction temperature. The connection of the thermocouple to any measuring device creates two more junctions at the termination point, and the actual electromotive force has to be corrected by a factor which is a function of the junction temperature (ambient or cold junction compensation). The K-type thermocouples are used in the unit, which could provide a 33 mV output at 8000°C, having an over-all temperature range of -200 and a tolerance of ±2.2°C.

Band heaters

Band heaters is a type of heater with long service life and superior heat transfer characteristics. The ceramic band heater is constructed of a series of interconnecting ceramic pieces in which a coiled alloy resistance wire is uniformly strung to form a flexible element. The element is covered with a thick fibre insulation and the top and sides are enclosed with a flexible stainless steel sheath.

Weight transducers

Weight transducer is sensed by a cantilever or platform load cell containing strain gauges in a Wheatstone bridge configuration. When weight is applied, it causes a slight distortion of the cell and produces a strain in the gauges which, leads to an electrical resistance change proportional to the weight. A constant excitation voltage is applied to the bridge so that the change in resistance can be measured by a change in the voltage of the output signal. Usually, the bridge resistance is adjusted so that the output signal is zero mill volts with no weight applied. The voltage level of the output signal will then be directly proportional to the weight.

8.3. Interface BOX (IBX) Description

The interface box is mainly a communication port between the process and the computer. The IBX also accepts power supply from the main power
station and converts it to different types of power supplies and distributes it to various components. The IBX accepts various types of process signals, converts those to digital signals and transmits them to the computer. The IBX also accepts digital signals from the computer, converts them to various analog controlling signals and transmits them to the process. The IBX controls process shut-down, independent of the computer, to protect the process during any malfunction. The following is a brief description of the various components

8.3.1. Power Relays

Power relays are used to interface the computer with high power components of the system (furnaces, motors, pumps, etc.). The relays for pumps and motors are standard mechanical contactor relays. The relays for electrical furnaces and heating tape are solid-state electronic relays of the zero-voltage switching type. The solid state relays are mounted on heat sinks for limiting the rise in temperature of the solid state switch by the energy dissipated in it and the control signal is derived from a digital I/O port of the control system.

8.3.2. Indicator Lights

The unit will have four indicating lights (blue, red, green, and red). In the state in which control power is being supplied before the master contactor, the blue light is illuminated. A push button green light indicates when system power is on with a coinciding push button red light indicating system power is off. By pressing either of these two, power can be manually turned off and turned on in conjunction with the computer power-on command. Finally, the red light is a flashing indication an alarm and a push-button for acknowledgement and silencing of the alarm horn.

8.4. Computer Description

The computer system accepts all of the process various signals, maintains a digital log of all the variables and also updates the signals. All the equipments will be controlled through the computer by simply ON/OFF
commands or by preprogrammed start-up commands. The computer control system will act automatically to maintain the system parameters.

8.5. Safety Management

The PEM Fuel Cell unit is designed to ensure the necessary safety requirements which are described below,

- Relief valves/rupture discs: Overpressure relief
- Safety switches (TSS, PSS): Hardwired shutdown
- High-high alarm (HIHI): Computer shutdown logic
- High alarm (HI): Computer alarm message
- Low alarm (LO): Computer alarm message
- Low-low alarm (LOLO): Computer shutdown logic

The process will be protected by overpressure by using various relief valves and/or rupture discs. The settings on these protection devices are based on the standard industry practices. A selected number of temperature safety switches will shut-down the entire system automatically in case of emergency. Four levels of alarm or shut-down signals are provided for each transmitting signal into the computer. A HIHI, HI, LO, LOLO, and bad sensor. The HI and LO alarms will be printed on the screen and printer and will switch on the alarm horn. The HIHI, LOLO and bad sensor alarms will, along with their logic, shut-down a part of the system and so protect the entire system. The shut-down logic of these alarms may not be manipulated by the operator.

8.6. Operation Procedure

The process will include four operating procedures, the heat-up, conditioning, regular operation and shut-down.
8.6.1. Heat-up

The stack needs to be brought to operating temperature 65°C before operation. Thus, the cooling/heating sub-system will heat up slowly PEM fuel cell stack with rate less than 10°C/min. The coolant flow should be greater than 250kg/h in order to ensure that homogeneous temperature distribution inside the stack. Finally, this phase is finished when the stack has reached the temperature of 60°C.

8.6.2. Conditioning

The conditioning step is really important to ensure optimal internal humidification of the membrane. During this phase the anode channel is feed with humidified N₂ at a flow rate 5Nl/min, while the cathode channel is feed with humidified air at flow 10Nl/min. Both gasses has to be humidified to 100 RH at 60°C and the stack should be flushed for at least 10 minutes.

8.6.3. Regular operation

During the regular operation the PEM fuel cell stack is feed with humidified hydrogen and air, and the fuel cell starts to produce electricity. The average cell voltage should be kept between 0.8V and 0.5V, while the minimum cell voltage should be above 0.3V. Operation with minimum cell voltage below the accepted range could cause serious damage to the stack. The operation procedure includes the following guidelines,

- Control average stack temperature at 65°C
- Activate hydrogen sensors in the system
- Set anode and cathode flow controllers to the minimum recommended flow
- Switch on the electronic load
- Set load to the PEM fuel cell stack

8.6.4. Shut down

The shut down procedure has to follow the guidelines in order to avoid any damage to the system.
• Switch off the load
• Switch off the air flow
• Purge anode channel with humidified $N_2$ until no $H_2$ is left
• Switch off tempering device

8.7. Modelling of the PEM Fuel Cell Stack (NEDSTACK 1kW)

The mathematical model presented in this section is one-dimensional in space and dynamic with respect to time. An energy balance is used to describe changes in the stack temperature which is assumed to be uniform. Multiple phases are considered where the liquid and gaseous water phases are in equilibrium. The gas phases are assumed to behave ideal as expressed by the ideal gas law. Also the dissolubility of the gases in the liquid water is neglected.

Regarding the fuel cell components the membrane and gas diffusion layer material is considered to be isotropic except for the porosity of the GDL which is affected by the liquid water content. The gas transport through the gas diffusion layers is only of diffusive nature; convective transport due to pressure gradients is neglected. Vapour is the only species able to pass the membrane, the pass through of all other species is neglected. The catalyst layer is modelled as the interface between membrane and gas diffusion layer and has no thickness. The chemical reaction between hydrogen and oxygen produces liquid water as the temperature is below the boiling temperature. The concentration is assumed to be uniform within a channel. Moreover the concentration losses at the cathode side are considered to be much larger than the anode side, hence the anodic concentration loss is neglected. Finally the electrochemistry constant electrode potentials throughout the stack are assumed.

8.7.1. Mathematical Model

The dynamic mathematical model presented in this section, investigate the performance of a 1kW PEM fuel cell. The model includes equations for the properties of the fuel and the air, such as density and viscosity. Moreover
includes the gas diffusion layer, membrane, channels and the electrochemical
equations. The nomenclature of the developed mathematical model is shown
in section 8.7.2.

In figure 8.3 the sign convention valid throughout the membrane elec-
trolyte assembly model is presented.

![Figure 8.3: Sign convention for streams within the membrane electrolyte assembly](image)

The flow of oxygen and hydrogen, the reactants, are defined as positive
towards the membrane whereas the flow of both liquid water and water
vapour is positive when directed off the membrane. Vapour crossing the
membrane is defined positive when flowing from the anode to the cathode
side of the fuel cell, while in the channels, flow is defined positive in the
direction from input to output position. The signs of the flow between
channel and membrane electrolyte assembly are defined according to figure
8.3.

**Physical Properties**

The saturation pressure is calculated as a function of the temperature, and
is given by equation (8.1) (Nam and Kaviany, 2003).

\[
p_{\text{sat}} = \exp \left\{ -\frac{5800}{T} + 1.391 - 0.4864T + 0.4176 \times 10^{-4}T^2 \\
- 0.1445 \times 10^{-7}T^3 + 6.546 \ln (T) \right\} \quad (8.1)
\]
Density of Liquid Water can be expressed as a function of the temperature by the following equation (Lide (2001)).

\[
\rho_l = c_1 + c_2 T + c_3 T^2 + c_4 T^3 + c_5 T^4 
\]

\[\text{(8.2)}\]

for \( T \in [10, 100] \)

\[
c_1 = 999.15 \\
c_2 = 0.141447 \\
c_3 = -0.0107106 \\
c_4 = 8.4069 \times 10^{-5} \\
c_5 = -3.19302 \times 10^{-7}
\]

The dynamic viscosity of liquid water depends on the temperature and pressure and can be calculated by using equation (8.3) as suggested by Likhachev (2003).

\[
\eta = \eta_0 \exp \left[ a p + \frac{E - b p}{R (T - \Theta - c p)} \right] 
\]

\[\text{(8.3)}\]

\[
E = 4.753 kJ/mol \\
\eta_0 = 2.4055 \times 10^{-5} Pa \ s \\
\Theta = 139.7 K \\
a = 4.42 \times 10^{-2} \text{bar} \\
b = 9.565 \times 10^{-4} kJ/mol \text{bar} \\
c = 1.24 \times 10^{-2} K/\text{bar}
\]

**Gas Diffusion Layer**

The gas diffusion layers are between the catalyst layers and the bipolar plates and distribute the reactants (fuel, air, water). Moreover, they allow reaction product water to exit the electrode surface and permit the passage between electrodes and flow channels. The mathematical modelling of the
gas diffusion layer is based on McKay et al. (2005) with the difference that is used distribution domains instead of a number of discrete sections.

**Diffusion of the Gaseous Species**

This section describes the transportation of the gaseous species within the gas diffusion layers. The equations are based on Fick’s Law.

The total pressure in the gas diffusion layer can be calculated as follows

\[
p_a(z) = p_{H_2}(z) + p_{N_2,a} + p_{v,a} \quad z \in [0, t_{GDL}] \quad (8.4a)
\]

\[
p_c(z) = p_{O_2}(z) + p_{N_2,c} + p_{v,c} \quad z \in [0, t_{GDL}] \quad (8.4b)
\]

A relation between concentration and pressure are based on the ideal gas law is given in equations (8.5).

\[
p_{i,a} = c_{i,a}RT \quad \text{for } i = v, H_2, N_2 \quad z \in [0, t_{GDL}] \quad (8.5a)
\]

\[
p_{i,c} = c_{i,c}RT \quad \text{for } i = v, O_2, N_2 \quad z \in [0, t_{GDL}] \quad (8.5b)
\]

The Fick’s Law of diffusion is used to calculate the molar fluxes depending on the concentration gradient and effective diffusion coefficient.

\[
N_{i,a}(z) = -D_{i,a}^e(z) \frac{\partial c_{i,a}}{\partial z} \quad \text{for } i = v, H_2 \quad z \in (0, t_{GDL}) \quad (8.6a)
\]

\[
N_{i,c}(z) = -D_{i,c}^e(z) \frac{\partial c_{i,c}}{\partial z} \quad \text{for } i = v, O_2 \quad z \in (0, t_{GDL}) \quad (8.6b)
\]

At the interface between membrane and gas diffusion layer the molar fluxes are equal to the chemical reaction taking place.

\[
N_{H_2,a}(0) = \frac{W_{H_2,rxn}}{A_{ac} \epsilon M_{H_2}} \quad (8.7a)
\]

\[
N_{v,a}(0) = -N_{v,mem} \quad (8.7b)
\]

\[
N_{O_2,c}(0) = \frac{W_{O_2,rxn}}{A_{ac} \epsilon M_{O_2}} \quad (8.7c)
\]

\[
N_{v,c}(0) = \frac{W_{H_2O,rxn}}{A_{ac} \epsilon M_{H_2O}} + N_{V,mem} \quad (8.7d)
\]
The equations at the boundary between gas diffusion layer and the channels are comparable to equations (8.6) with the differences that the effective diffusion coefficient takes the last value within the GDL \((t_{GDL})\) and not the channel value.

The effective diffusion coefficients are functions of the corresponding species diffusion coefficient, the porosity of the GDL’s material and the saturation ratio with liquid water as can be seen from equations (8.8). The liquid water saturation ratio is defined in (8.17) in the following part dealing with the transport of liquid water.

\[
D_{i,a}^e(z) = D_i \epsilon_a(z) \left( \frac{\epsilon_a(z) - 0.11}{1 - 0.11} \right)^{0.785} (1 - s_a(z))^2 \quad \text{for } i = v, H_2 \quad z \in [0, t_{GDL}]
\]

\[
D_{i,c}^e(z) = D_i \epsilon_c(z) \left( \frac{\epsilon_c(z) - 0.11}{1 - 0.11} \right)^{0.785} (1 - s_c(z))^2 \quad \text{for } i = v, O_2 \quad z \in [0, t_{GDL}]
\]

Taking into account the amount of liquid volume present within the gas diffusion layer, the porosity is defined as the ratio of the void volume to the total volume.

\[
\epsilon_k(z) = \frac{V_{pores} - V_{liq,k}(z)}{V_{GDL}} \quad \text{for } k = a, c z \in [0, t_{GDL}]
\]

The porosity of the channel is defined as

\[
\epsilon_k(t_{GDL}) = \frac{V_{ch,k}}{V_{ch,k} + V_{liq,k}(t_{GDL})} \quad \text{for } k = a, c
\]

The species concentration depends on the incoming and outgoing fluxes of the control volume as well as possible source or sink terms like condensation.
and evaporation.

\[
\begin{align*}
\frac{dc_{H_2}(z)}{dt} &= -\frac{\partial N_{H_2}}{\partial z} \bigg|_z \quad z \in (0, t_{GDL}) \quad (8.11a) \\
\frac{dc_{O_2}(z)}{dt} &= -\frac{\partial N_{O_2}}{\partial z} \bigg|_z \quad z \in (0, t_{GDL}) \quad (8.11b) \\
\frac{dc_{v,i}(z)}{dt} &= -\frac{\partial N_{v,i}}{\partial z} \bigg|_z + R_{\text{evap},i}(z) \quad \text{for } i = a, c \quad z \in (0, t_{GDL}) \quad (8.11c)
\end{align*}
\]

At the boundary between the membrane equations (8.11) remain valid. The concentrations on the channels, i.e. \(c_i(t_{GDL})\), can be calculated by the mass balances in the channels (equations (8.39) and (8.42a)). To convert from mass to concentration the following equation is used

\[
c_{i,k}(t_{GDL}) = \frac{m_{i,k}(\text{channel})}{M_i V_{ch,k}} \quad \text{for } \begin{cases} k = a & i = H_2, v \\ k = c & i = O_2, v \end{cases} \quad (8.12)
\]

It is assumed that the concentration of nitrogen is constant throughout the gas diffusion layer and has the same value as in the channels.

\[
c_{N_2,k}(z) = c_{N_2,k}(\text{channel}) \quad \text{for } k = a, c \quad z \in [0, t_{GDL}] \quad (8.13)
\]

**Liquid Water Capillary Transport**

The volume of liquid water in the GDL is calculated through the capillary liquid water flow, \(W_{\text{liq},k}\), and the evaporation rate, \(R_{\text{evap}}\)

\[
\rho_{\text{liq}} \frac{dV_{\text{liq},k}(z)}{dt} = W_{\text{liq},k}(z) - W_{\text{liq},k}(z + 1) - \min \left\{ R_{\text{evap}} V_{\text{v}}, \frac{V_{\text{liq},i}}{M_{\text{v}}} \right\} \quad \text{for } k = a, c \quad z \in [0, t_{GDL}] \quad (8.14)
\]

The minimum is to ensure that the amount of liquid evaporating cannot be more than what is present in liquid state.

The liquid volume in the surface of both channels is related to the mass balance of water by

\[
V_{\text{liq},k}(GDL\text{thickness}) = \frac{m_{\text{liq},k}(\text{channel})}{\rho_{\text{liq}}} \quad \text{for } k = a, c \quad (8.15)
\]
During the normal operation of the unit, the pore fills with liquid water, which increases the capillary pressure and causes the water to flow. This phenomenon causes the liquid water to flow through the GDL, resulting in the injection of water in the channel. The general equation for the capillary liquid flow is

$$W_{\text{liq}} = -\frac{A_{\text{GDL}} \rho_{\text{liq}} K_{\text{rel}}}{\mu_{\text{liq}}} \left( \frac{dp_c}{dS} \right) \left| \frac{\partial S}{\partial z} \right| \quad (8.16)$$

where $K_{\text{rel},k}(z) = S_k(z)^3$ is the relative permeability and $S_k(z)$ is the reduced liquid saturation,

$$s_k(z) = \frac{V_{\text{liq},k}(z)}{V_p} \quad \text{for} \ k = a, c \quad z \in [0, t_{\text{GDL}}) \quad (8.17)$$

The capillary pressure expresses the surface tension of the water droplets and can be calculated by the Leverette J-function

$$p_{c,k}(z) = \frac{\sigma \cos \Theta_c}{(K/\epsilon_k(z))^{0.5}} \left[ 1.417 S_k(z) - 2.12 S_k(z)^2 + 1.263 S_k(z)^3 \right] \quad (8.19)$$

for $k = a, c \quad z \in [0, t_{\text{GDL}}]$ (8.19)

The contact angle of a water droplet in a diffusion medium is about 120°, which results in a value for the absolute value of the cosine of 0.5 Nam and Kaviany (2003).

Substituting the above equations into equation (8.16) results in the fol-
lowing modified equation for the capillary liquid flow

\[ W_{\text{liq},k}(z + 1) = -\frac{A_{ac}\rho_{\text{liq}}K_{rel,k}(z)}{\mu_{\text{liq}}} \times \sigma |\cos \Theta_c| \sqrt{K_{c}(x)^3} \times \]
\[ \left[ 1.417 - 4.240S_k(z) + 3.789S_k(z)^2 \right] \frac{\partial S_k}{\partial z} \bigg|_z \]

(8.20)

for \( k = a, c \quad z \in (0, t_{\text{GDL}}) \)

The flow of liquid water between the GDL and the membrane is zero at the anode since no liquid water can pass the membrane, while at the cathode side is positive since the reaction is taking place. Depending on whether the product is liquid water or water vapour there is a liquid flow or not. The situation at the boundaries towards the membrane are expressed mathematically as

\[ W_{\text{liq},a}(0) = 0 \quad (8.21a) \]
\[ W_{\text{liq},c}(0) = \begin{cases} 0 & \text{if vapour is the reaction product} \\ W_{\text{liq,rxn}} & \text{if liquid water is the reaction product} \end{cases} \quad (8.21b) \]

The molar evaporation rate in the gas diffusion layer is given by

\[ R_{\text{evap},k}(z) = \begin{cases} \frac{\gamma_{p_{\text{sat}}-p_{\text{v},k}(z)}}{RT} & \text{for } s > 0 \\ \min \left( 0, \frac{\gamma_{p_{\text{sat}}-p_{\text{v},k}(z)}}{RT} \right) & \text{for } s \leq 0 \end{cases} \quad (8.22) \]

for \( k = a, c \quad z \in [0, t_{\text{GDL}}] \)

**Membrane**

The vapour flow across the membrane depends on the diffusive and convective fluxes as well as the electro-osmotic drag.

\[ N_{v,\text{mem}} = N_{v,\text{diff}} + N_{v,\text{conv}} + N_{v,\text{drag}} \quad (8.23) \]

The concentration gradient is expressed by a single step linear difference
as the thickness of the membrane is very small 38.1µm.

\[
N_{i,\text{diff}} = n_{\text{diff}} D_{mb} \left( \lambda_c - \lambda_a \right) \frac{\rho_{mb,\text{dry}}}{M_{mb,\text{dry}} t_{mb}} \tag{8.24}
\]

\[
c_w = \frac{\rho_{mb,\text{dry}}}{M_{mb,\text{dry}}} \lambda \tag{8.25}
\]

The water content depends on the water activity as shown in equation (8.27). Springer et al. (1991a) estimate the value of the water content to be 16.8 while McKay et al. uses 16.0.

\[
\lambda_i = \begin{cases} 
0.043 + 17.81a_i^2 + 36.0a_i^3 & \text{for } 0 < a_i \leq 1 \\
14.0 + 1.4 (a_i - 1) & \text{for } 1 < a_i \leq 3 \\
16.8 & \text{elsewhere}
\end{cases} \tag{8.26}
\]

for \( i = a, c, mb \) \tag{8.27}

The Water activity is defined as the ratio of the partial vapour pressure to the saturation pressure at a given temperature Garbosa-Canovas et al. (2007). Equation (8.28) gives the mathematical expression for the water activity at either side of the membrane.

\[
a_i = \frac{p_{v,i}(0)}{p_{\text{sat}}} \quad \text{for } i = a, c \tag{8.28}
\]

The membrane water activity is the mean of of the water activities on the anode and cathode boundary of the membrane.

\[
a_{mb} = \frac{a_a + a_c}{2} \tag{8.29}
\]

The diffusion coefficient of the vapour across the membrane \( D_{v,mb} \) is calculated using corrected diffusion coefficient \( D_{\lambda} \) which varies piecewise with
the water content of the membrane (Dutta et al. (2001)).

\[ D_{v,mb} = D_\lambda \exp \left( 2416 \left( \frac{1}{303} - \frac{1}{T_{cell}} \right) \right) \]  
(8.30)

\[ D_\lambda = \begin{cases} 
10^{-10} & \text{for } \lambda < 2 \\
10^{-10} (1 + 2 (\lambda - 2)) & \text{for } 2 \leq \lambda \leq 3 \\
10^{-10} (3 - 1.67 (\lambda - 3)) & \text{for } 3 < \lambda < 4.5 \\
1.25 \times 10^{-10} & \text{for } \lambda \geq 4.5 
\end{cases} \]  
(8.31)

The vapour molar flux caused by convection is expressed as

\[ N_{v,conv} = -\left( \frac{c_{w,a} + c_{w,c}}{2} \right) \frac{k_p}{\mu} \left( p_{v,c}(0) - p_{v,a}(0) \right) t_{mb} \]  
(8.32)

where the pressure gradient is approximated by a single step linear difference. The values for the hydraulic permeability and the pore-water viscosity are taken from Yi and Nguyen (1998), and similar values are presented in Bernardi and Verbrugge (1992) and Ge and Yi (2003).

The electro-osmotic drag can be calculated by using the following equation

\[ N_{v,drag} = -n_{drag} \frac{i}{F} = -n_{drag} \frac{I}{A_{active} F} \]  
(8.33)

The drag coefficient can be calculated by equation (8.34) as Dutta (Springer et al. (1991a)) proposed.

\[ n_{drag} = 0.0029\lambda_{mb}^2 + 0.05\lambda_{mb} - 3.4 \times 10^{-19} \]  
(8.34)

**Channel**

The anode/cathode channel mathematical model is based on the first principles and the mass balances. The behaviour of the gas streams in the anode and cathode channel is similar, thus they are presented together rather than separated in anode and cathode equations. The developed mathematical model for the channels is based on the work of McKay et al. (2005). In this work the individual fuel cells are connected using the mathematical relationship between the channels and the gas diffusion layers, namely the species flow between them as described below by equations (8.47).

The humidity value depends on the amount of water and can be described
by the following equation

\[
\phi_{k,\text{channel}} = \min \left\{ \frac{1}{\frac{m_{w,k}(\text{channel})RT}{M_{w}V_{ch,k}}} \right\} \quad \text{for } k = a, c \quad (8.35)
\]

The partial vapour pressure is calculated using the relative humidity and the saturation pressure.

\[
p_{v,k}(t_{\text{GDL}}) = \phi_{k}(\text{channel})p_{\text{sat}} \quad \text{for } k = a, c \quad (8.36)
\]

The partial pressure of the gas components in the channel can be expressed by using the ideal gas law

\[
p_{i,a}(t_{\text{GDL}}) = \frac{m_{i,a}(\text{channel})RT}{M_{i}(V_{ch,a} - V_{liq,a}(t_{\text{GDL}}))} \quad \text{for } i = H_2, N_2, v \quad (8.37a)
\]

\[
p_{i,c}(t_{\text{GDL}}) = \frac{m_{i,c}(\text{channel})RT}{M_{i}(V_{ch,c} - V_{liq,c}(t_{\text{GDL}}))} \quad \text{for } i = O_2, N_2, v \quad (8.37b)
\]

The amount of vapour water as presented is the following equation. The amount of vapour water cannot be more than the total mass of the water (liquid and vapour) in the channel.

\[
m_{v,k}(\text{channel}) = \min \left\{ \frac{m_{w,k}(\text{channel})}{\frac{p_{\text{sat}}(V_{ch,k} - V_{liq,k}(t_{\text{GDL}}))RT}{M_{w}}} \right\} \quad \text{for } k = a, c \quad (8.38)
\]

The mass balance of each component is computed in equation (8.39) and depends on incoming and outgoing streams. The mass balance for the water, in both liquid and gas state, is treated in equation (8.42a) in combination with the equations (8.35) and (8.38). The liquid water is only present when the saturation pressure is above the vapour pressure.

\[
\frac{dm_{i,k}(\text{channel})}{dt} = W_{i,k,in} - W_{i,k,out} + n_{\text{cells}}W_{i,k,\text{GDL}}
\]

\[
\text{for } i = H_2, N_2 \text{ and } k = a \quad (8.39a)
\]

\[
i = O_2, N_2 \text{ and } k = c \quad (8.39b)
\]
\[
\frac{dm_{w,k}(\text{channel})}{dt} = W_{v,k,\text{in}} - W_{v,k,\text{out}} - W_{liq,k,\text{out}} + n_{cell} W_{w,k,\text{GDL}} \quad (8.40)
\]

for \( k = a, c \)

\[
m_{liq,k}(\text{channel}) = \begin{cases} 
    m_{w,k}(\text{channel}) - m_{v,k}(\text{channel}) & \text{for } p_{v,k} > p_{\text{sat}} \\
    0 & \text{else} 
\end{cases} \quad (8.41)
\]

for \( k = a, c \)

The inlet mass flow rates are calculated using the total inlet stream and the individual mass fractions as follows

\[
W_{i,a,\text{in}} = x_{i,a,\text{in}} W_{a,\text{in}} \quad \text{for } i = H_2, N_2, v \quad (8.42a)
\]

\[
W_{i,c,\text{in}} = x_{i,c,\text{in}} W_{c,\text{in}} \quad \text{for } i = O_2, N_2, v \quad (8.42b)
\]

The total outlet mass flow rate depends on the pressure difference between the channel and outlet pressure under the assumption that the composition of the outlet stream is identical to the composition inside the channel.

\[
W_{a,\text{out}} = k_a \left( \sum_i^n p_{i,a}(\text{channel}) - p_{a,\text{out}} \right) \quad \text{for } n = H_2, N_2, v \quad (8.43)
\]

\[
W_{i,a,\text{out}} = \frac{m_{i,a}(\text{channel})}{\sum_j^n m_{j,a}(\text{channel})} W_{a,\text{out}} \quad \text{for } i = H_2, N_2, v \quad n = H_2, N_2, v \quad (8.44)
\]

\[
W_{c,\text{out}} = k_a \left( \sum_i^n p_{i,c}(\text{channel}) - p_{c,\text{out}} \right) \quad \text{for } n = O_2, N_2, v \quad (8.45)
\]

\[
W_{i,c,\text{out}} = \frac{m_{i,c}(\text{channel})}{\sum_j^n m_{j,c}(\text{channel})} W_{c,\text{out}} \quad \text{for } i = O_2, N_2, v \quad n = O_2, N_2, v \quad (8.46)
\]

The mass flow rates diffusing from the channel into the GDL are given in
equations (8.47).

\[ W_{i,k,GDL} = N_{i,k}(t_{GDL})M_iA_{active} \epsilon \]

for \( i = H_2, N_2, v \) and \( k = a \) \hspace{1cm} (8.47a)

\[ i = O_2, N_2, v \) and \( k = c \) \hspace{1cm} (8.47b)

\[ W_{w,k,GDL} = W_{liq,k,GDL} + W_{v,k,GDL} \]

for \( k = a, c \) \hspace{1cm} (8.47c)

\[ W_{liq,k,GDL} = W_{liq,k} \]

for \( k = a, c \) \hspace{1cm} (8.47d)

Inlet Streams

The properties of the inlet streams depend both on the electrode to which they are fed as well as the operation type. The cathode is only supplied with humidified air during the whole operation cycle whereas for the anode inlet humidified hydrogen as well as humidified nitrogen need to be considered during the start-up process.

Anode Inlet Stream

At the anode the inlet stream during start up and operation consists of humidified hydrogen. Before and after the fuel cell is operated it is purged with nitrogen.

The humidity ratio depends on the relative humidity as well as the molecular weight of vapour and the dry gas mixture. Based on the definition of the humidity ratio it can be calculated as shown in the equation below.

\[ \omega_{in,a} = \frac{M_v\phi_{in,a}p_{sat}}{p_{in,a} - \phi_{in,a}p_{sat}} \left( \frac{y_{H_2,in,a}}{M_{H_2}} + \frac{y_{N_2,in,a}}{M_{N_2}} \right) \]

(8.48)

Depending on the total mass flow rate, the mass fractions of the components hydrogen, nitrogen and vapour are expressed by

\[ y_{i,in,a} = \frac{W_{i,in,a}}{\sum W_{i,in,a}} \quad \text{for} \quad i = H_2, N_2, v \]

(8.49)

The individual mass flow rates of hydrogen, nitrogen and vapour are
calculated using the following equations:

\[ W_{H_2} = \kappa \frac{1}{1 + \omega} W_{\Sigma, in, a} \quad (8.50) \]
\[ W_{N_2} = (1 - \kappa) \frac{1}{1 + \omega} W_{\Sigma, in, a} \quad (8.51) \]
\[ W_{H_2} = \frac{\omega}{1 + \omega} W_{\Sigma, in, a} \quad (8.52) \]

where \( \kappa \) is a binary variable with value 0 when nitrogen is the inlet gas and 1 when hydrogen is supplied to the anode channel, since the inlet stream either consists of humidified hydrogen or nitrogen but never nitrogen and hydrogen the same time.

**Cathode Inlet Stream**

The inlet stream at the cathode side always consists of humidified air with more than 70% humidity ratio.

Air as considered within the fuel cell consists of 0.21% oxygen and 0.79% nitrogen. The mass fraction of oxygen can be converted to a mole fraction by using equation (8.53).

\[ y_{O_2, air} = \frac{x_{O_2, air} M_{O_2}}{x_{O_2, air} M_{O_2} + (1 - x_{O_2, air}) M_{N_2}} \quad (8.53) \]

The humidity ratio within the cathode inlet stream depends on the relative humidity and the properties of vapour and dry air.

\[ \omega_{in,c} = \frac{M_v \phi_{in,c} p_{sat}}{M_{dry \ air} (p_{in,c} - \phi_{in,c} p_{sat})} \quad (8.54) \]

Assuming only vapour, oxygen and nitrogen in the gaseous phase, the mass flow rates of the components can be expressed using the following equations

\[ W_{O_2, in, c} = y_{O_2, air} \frac{1}{1 + \omega_{in,c}} W_{\Sigma, in, c} \quad (8.55) \]
\[ W_{N_2, in, c} = (1 - y_{O_2, air}) \frac{1}{1 + \omega_{in,c}} W_{\Sigma, in, c} \quad (8.56) \]
\[ W_{H_2, in, c} = \frac{\omega}{1 + \omega_{in,c}} W_{\Sigma, in, c} \quad (8.57) \]
Electrochemistry

The chemical reaction that takes place during the operation of the polymer electrolyte membrane (PEM) fuel cell is the oxidation of hydrogen. The two reactions, oxidation and reduction, are separated by the membrane which only allows protons and vapour to pass. To complete the reaction electrons move in an external circuit from the anode to the cathode and by doing so produce an electrical current. Equations (8.58) to (8.60) can be used to calculate the mass flowrate of the reaction in a single cell.

\[ W_{H_2, rxn} = -\frac{IM_{H_2}}{2F} \quad (8.58) \]

\[ W_{O_2, rxn} = -\frac{IM_{O_2}}{4F} \quad (8.59) \]

\[ W_{H_2O, rxn} = \frac{IM_{H_2O}}{2F} \quad (8.60) \]

The actual supply of hydrogen and oxygen is larger than what is stochastically desired for the reaction. The ratio between the amount of reactants supplied to the stack and the amount needed for the reaction (excess ratio) can be computed as follows

\[ \Psi_i = \frac{W_{i, in}}{W_{i, rxn}} \times \text{ncells} \] for \( i = H_2, O_2 \) \quad (8.61)

Usually, the excess ratio has the value of \( \Psi_{H_2} = 1.25 \) and \( \Psi_{O_2} = 2 \).

The fuel cell voltage consists of the thermodynamic equilibrium potential, the Nernst potential, and the three voltage losses due to the electrochemical reaction, ohmic losses accompanying proton and electron conduction as well as the concentration losses caused by mass transport limitations.

\[ E_{\text{Stack}} = n_{\text{cells}} \left( E_{\text{Nernst}} - \eta_{\text{act}} - \eta_{\text{ohm}} - \eta_{\text{conc}} \right) \quad (8.62) \]

The Nernst voltage depends on the partial pressure and temperature, and for the case of PEM fuel cell reaction \( 2H_2(g) + O_2(g) \rightleftharpoons 2H_2O(l) \) the following equation is suggested by Amphlett et al. (1995).

\[ E_{\text{Nernst}} = 1.229 - 8.5 \times 10^{-4} (T - 298.15) + \frac{RT}{2F} \ln \left[ \frac{p_{H_2}}{p_{O_2}^{0.5}} \right] \quad (8.63) \]

Activation loss or activation overvoltage occurs because of the need to move
electrons and to break chemical bonds in the anode and cathode. Part of available energy is lost in driving chemical reactions that transfers electrons to and from electrodes. However, the oxidation of $H_2$ at the anode is very rapid and the reduction of $O_2$ at the cathode is slow, then the voltage drop due to activation loss is dominated by the cathode reaction. Ohmic loss arises from resistance of polymer membrane to the transfer of proton and the resistance of the electrode and collector plate to the transfer of electron. Resistance depends on the membrane humidity and cell temperature.

Amphlett et al. (1995) proposed the following parametric expressions for the activation loss and the ohmic loss as shown in equations (8.64) and (8.65) where the $\zeta$ represents constant parameters.

$$\eta_{act} = \zeta_1 + \zeta_2 T + \zeta_3 T (\ln I) + \zeta_4 T (\ln C_{O_2}^* )$$  (8.64)

$$\eta_{ohm} = I (\zeta_5 + \zeta_6 T + \zeta_7 I)$$  (8.65)

Finally concentration loss results from the drop in concentration of the reactants as they are consumed in the reaction. This explains the rapid voltage drop at high current density (O’Hayre et al. (2009)).

$$\eta_{conc} = \frac{RT}{4F} \left( 1 + \frac{1}{0.5} \right) \ln \left( \frac{I_L}{I_L - I} \right)$$  (8.66)

The limiting current is expressed in the following way

$$I_L = 2FA_{ac} \times \min \left( D_{O_2,c}(z) \right) \times \frac{c_{O_2,c}(t_{GDL})}{t_{GDL}}$$  (8.67)

The concentration losses can also be expressed as a function of the partial pressures of the reactants inside the fuel cell.

$$\eta_{conc} = \frac{RT}{2F} \left( 1 + \frac{1}{0.5} \right) \ln \left( \frac{p_{H_2}(0) p_{O_2}(0)^{1/2}}{p_{H_2}(t_{GDL}) p_{O_2}(t_{GDL})^{1/2}} \right)$$  (8.68)

**Thermodynamic balance**

The heat and temperature subsystem includes the fuel cell stack cooling system and the reactant temperature system. The reaction is highly exothermic, so that as current is drawn by the traction motor, a large quantity of heat is generated within the fuel cell. A passive dissipation by air convection
and radiation through the external surface of the stack would not be sufficient and a cooling is required (air or water). The developed mathematical model for the energy balance of the fuel cell is based on the work of Del Real et al. (2007) and assumes that fuel energy that is not converted into electrical energy becomes heat. Final the model assumes uniform temperature inside the fuel cell, since we don’t have information about the geometry inside the fuel cell.

\[
m_{\text{Stack}}c_{\text{Stack}} \frac{dT}{dt} = \dot{H}_{\text{in}} - \dot{H}_{\text{out}} + \dot{Q}_{\text{rxn}} + \dot{Q}_{\text{gen}} - \dot{Q}_{\text{rad}} - \dot{Q}_{\text{cool}} \quad (8.69)
\]

For the enthalpy of the inlet and outlet streams the following equations are used

\[
\dot{H}_{a,d} = \sum_{i} W_{i,a,d} h_{i} \quad \text{for} \quad d=\text{in, out} \quad i = H_{2}, N_{2}, v \quad (8.70)
\]

\[
\dot{H}_{c,d} = \sum_{i} W_{i,c,d} h_{i} \quad \text{for} \quad d=\text{in, out} \quad i = O_{2}, N_{2}, v \quad (8.71)
\]

The heat of the reaction can be expressed as follows

\[
\dot{Q}_{\text{rxn}} = -W_{H_{2}O,\text{rxn}} \Delta h_{f,\text{liq}}^{\text{o}} \quad (8.72)
\]

The heat generated is equal to the electrical power of the cell since the energy conversion from chemical to electrical energy also produces heat as a by-product.

\[
\dot{Q}_{\text{gen}} = P_{\text{Stack}} = I \times E_{\text{Stack}} \quad (8.73)
\]

The amount of heat radiated by the stack’s surface is a function of the surface area and the difference between stack and ambient temperature as well as the Boltzmann constant and an effectiveness factor of the radiation.

\[
\dot{Q}_{\text{rad}} = \epsilon_{\text{rad}} \sigma A_{\text{surface}} (T^{4} - T_{\text{amb}}^{4}) \quad (8.74)
\]

Finally the energy removed the fuel cell from the coolant can be expressed as follows

\[
\dot{Q}_{\text{cool}} = \epsilon_{\text{cool}} W_{\text{cool}} c_{p,\text{cool}} (T - T_{\text{cool}}) \quad (8.75)
\]

where \(\epsilon_{\text{cool}}\) is an effectiveness factor, \(W_{\text{cool}}\) the mass flow rate of the
coolant, $c_p$ the coolant’s heat capacity and $T_{cool}$ its temperature.
8.7.2. Nomenclature

**Latin Characters**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>water activity</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>area</td>
<td>m²</td>
</tr>
<tr>
<td>c</td>
<td>molar concentration</td>
<td>mol/m²</td>
</tr>
<tr>
<td>cp</td>
<td>specific heat capacity</td>
<td>J/kg K</td>
</tr>
<tr>
<td>cStack</td>
<td>specific heat capacity of a fuel cell stack</td>
<td>J/kg K</td>
</tr>
<tr>
<td>D</td>
<td>diffusion coefficient</td>
<td>m²/s</td>
</tr>
<tr>
<td>E</td>
<td>potential</td>
<td>V</td>
</tr>
<tr>
<td>F</td>
<td>Faraday constant</td>
<td>C/mol</td>
</tr>
<tr>
<td>h</td>
<td>mass specific enthalpy</td>
<td>J/kg</td>
</tr>
<tr>
<td>(h_{f,liq}^\circ)</td>
<td>standard enthalpy of formation of liquid water</td>
<td>J/kg</td>
</tr>
<tr>
<td>H</td>
<td>enthalpy flow rate</td>
<td>W</td>
</tr>
<tr>
<td>i</td>
<td>current density</td>
<td>A/m²</td>
</tr>
<tr>
<td>I</td>
<td>current</td>
<td>A</td>
</tr>
<tr>
<td>k</td>
<td>orifice constant</td>
<td>m s</td>
</tr>
<tr>
<td>K</td>
<td>absolute permeability</td>
<td>m²</td>
</tr>
<tr>
<td>K_rel</td>
<td>relative permeability of liquid water</td>
<td>-</td>
</tr>
<tr>
<td>m</td>
<td>mass</td>
<td>kg</td>
</tr>
<tr>
<td>M</td>
<td>molecular weight</td>
<td>kg/mol</td>
</tr>
<tr>
<td>n</td>
<td>number of moles</td>
<td>mol</td>
</tr>
<tr>
<td>(n_{cells})</td>
<td>number of cells</td>
<td>-</td>
</tr>
<tr>
<td>N</td>
<td>molar flux</td>
<td>mol/s m²</td>
</tr>
<tr>
<td>p</td>
<td>pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>P</td>
<td>power</td>
<td>W</td>
</tr>
<tr>
<td>(\dot{Q})</td>
<td>heat flow rate</td>
<td>W</td>
</tr>
<tr>
<td>R</td>
<td>ideal gas constant</td>
<td>J/mol K</td>
</tr>
<tr>
<td>(R_{evap})</td>
<td>evaporation rate</td>
<td>mol/s m²</td>
</tr>
<tr>
<td>s</td>
<td>volume fraction of liquid water to the total volume</td>
<td>-</td>
</tr>
<tr>
<td>S</td>
<td>reduced liquid water saturation</td>
<td>-</td>
</tr>
<tr>
<td>t</td>
<td>thickness</td>
<td>m</td>
</tr>
<tr>
<td>T</td>
<td>temperature</td>
<td>K</td>
</tr>
<tr>
<td>V</td>
<td>volume</td>
<td>m³</td>
</tr>
<tr>
<td>W</td>
<td>mass flow rate</td>
<td>kg/s</td>
</tr>
<tr>
<td>x</td>
<td>molar ratio</td>
<td>-</td>
</tr>
<tr>
<td>y</td>
<td>mass ratio</td>
<td>-</td>
</tr>
<tr>
<td>z</td>
<td>direction across the gas diffusion layer</td>
<td>m</td>
</tr>
</tbody>
</table>

195
<table>
<thead>
<tr>
<th>Greek Character</th>
<th>Meaning</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>transfer coefficient</td>
<td></td>
</tr>
<tr>
<td>$\gamma$</td>
<td>volumetric condensation coefficient</td>
<td>$s^{-1}$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>thickness</td>
<td>m</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>porosity</td>
<td></td>
</tr>
<tr>
<td>$\epsilon_{cool}$</td>
<td>cooling effectiveness factor</td>
<td></td>
</tr>
<tr>
<td>$\epsilon_{rad}$</td>
<td>radiation effectiveness factor</td>
<td></td>
</tr>
<tr>
<td>$\zeta$</td>
<td>coefficient in voltage loss equations</td>
<td></td>
</tr>
<tr>
<td>$\eta$</td>
<td>overpotential</td>
<td>V</td>
</tr>
<tr>
<td>$\Theta_c$</td>
<td>contact angle</td>
<td></td>
</tr>
<tr>
<td>$\kappa$</td>
<td>zero one variable for hydrogen inlet stream</td>
<td></td>
</tr>
<tr>
<td>$\lambda$</td>
<td>water content</td>
<td></td>
</tr>
<tr>
<td>$\mu$</td>
<td>viscosity</td>
<td>$kg/m s$</td>
</tr>
<tr>
<td>$\xi$</td>
<td>stoichiometric coefficient</td>
<td></td>
</tr>
<tr>
<td>$\rho$</td>
<td>density</td>
<td>$kg/m^3$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Stefan-Boltzmann constant</td>
<td>$W/m^2 K^4$</td>
</tr>
<tr>
<td>$\sigma_{303K}$</td>
<td>membrane conductivity at 303K</td>
<td>$\Omega^{-1}m^{-1}$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>relative humidity</td>
<td></td>
</tr>
<tr>
<td>$\Psi$</td>
<td>excess ratio</td>
<td></td>
</tr>
<tr>
<td>$\omega$</td>
<td>humidity ratio</td>
<td></td>
</tr>
</tbody>
</table>
Subscripts

a anode
dact activation
damb ambient
dc cathode
dcell single fuel cell related
dch channel
dconc concentration
dconv convection
dcool coolant
ddiff diffusion
ddrag electro-osmotic drag
dg gaseous state

gDGL gas diffusion layer
ggen generation
H2 hydrogen
H2O water
din inlet
dl liquid state
dL limiting
dliq liquid water

dmb membrane
N2 Nitrogen
O2 Oxygen
dohm ohmic
dout outlet
drad radiation
drxn reaction
dsat saturation
dv vapour

Superscripts

dene effective
d* at the reaction site
deo channel value
8.7.3. Parameter Values

Due to the lack of information about the design of NedStack PEM fuel cell some of the parameters are adopted by McKay et al. (2005) and in the future will be estimated from the experimental data.

Gas Diffusion Layer

The parameter values which have been used for the GDL model is given in the following table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{GDL}$</td>
<td>210µm</td>
</tr>
<tr>
<td>$\epsilon_0$</td>
<td>0.5</td>
</tr>
<tr>
<td>$K$</td>
<td>$2.55 \times 10^{-13} m^2$</td>
</tr>
<tr>
<td>$s_{im}$</td>
<td>0.1</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>900</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.0644 N/m</td>
</tr>
<tr>
<td>$\Theta_c$</td>
<td>120$^\circ$</td>
</tr>
<tr>
<td>$D_{H_2}$</td>
<td>$1.14 \times 10^{-4} m^2/s$</td>
</tr>
<tr>
<td>$D_{O_2}$</td>
<td>$3.45 \times 10^{-5} m^2/s$</td>
</tr>
<tr>
<td>$D_v$</td>
<td>$3.03 \times 10^{-5} m^2/s$</td>
</tr>
</tbody>
</table>

Table 8.2: Parameter values of the gas diffusion layer as used in the model

Membrane

The properties of the membrane are based on the work of McKay et al. (2005) except for the water permeability and the water pore viscosity (Yi and Nguyen (1998)).

Stack Properties

Mass and surface area of the fuel cell stack are part of the specifications given by NedStack as well as the number of individual cells inside the stack. Values for the channel volumes and the orifice constants are estimated and compared with the work of McKay et al. (2005)

Energy Balance

Table 8.5 presents the parameters used for the heat balances of the system.
### Table 8.3: Parameter values of the membrane as used in the model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{mb}$</td>
<td>38.1 $\mu$m</td>
</tr>
<tr>
<td>$\rho_{mb, dry}$</td>
<td>$1.9 \times 10^3$ kg/m$^3$</td>
</tr>
<tr>
<td>$M_{mb, dry}$</td>
<td>1 kg/mol</td>
</tr>
<tr>
<td>$k_p$</td>
<td>$1.58 \times 10^{-18}$ m$^2$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>$3.565 \times 10^{-4}$ kg/m s</td>
</tr>
</tbody>
</table>

### Table 8.4: Parameter values of the fuel cell stack as used in the model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{surface}$</td>
<td>0.3326 m$^2$</td>
</tr>
<tr>
<td>$m_{Stack}$</td>
<td>15 kg</td>
</tr>
<tr>
<td>$n$</td>
<td>8</td>
</tr>
<tr>
<td>$V_{ch,a}$</td>
<td>$4.3 \times 10^{-4}$ m$^3$</td>
</tr>
<tr>
<td>$k_a$</td>
<td>$9.344 \times 10^{-7}$ kg/s Pa</td>
</tr>
<tr>
<td>$V_{ch,c}$</td>
<td>$3.8 \times 10^{-4}$ m$^3$</td>
</tr>
<tr>
<td>$k_c$</td>
<td>$11.31 \times 10^{-7}$ kg/s Pa</td>
</tr>
</tbody>
</table>

#### 8.7.4. Model Validation

The developed mathematical model has been validated through the experimental data provided by the manufacturer, since there weren’t experimental data of our experimental unit. The data include information for the polarization curve at the nominal operation conditions, which are used to estimate the electrochemical parameters as well as the GDL thickness, which is a critical factor to the limited current. The parameters $\zeta$ in equations (8.64) and (8.65) are estimated using gPROMS. The constant relative variance model is used with the boundary values 0.01 and 0.5. As initial guess a value of 0.2 is used. The concentration losses can be calculated by two equations, where the first model that has been used (equation (8.66)) calculates the loses based on the limiting current. The second model evaluates the concentration loss using the pressure values of hydrogen and water at the reaction sites (equation (8.68)). Also three different values for the gas diffusion layer thickness are considered, namely $200 \mu m$, $205 \mu m$ and $209.75 \mu m$, and the thickness of the gas diffusion layer was determined to be around $205 \mu m$. 

199
Parameter | Value
--- | ---
$T_{cool}$ | 62.5$^\circ$C
$W_{cool}$ | 250kg/h
$\epsilon_{cool}$ | 0.5
$T_{amb}$ | 25$^\circ$C
$\epsilon_{rad}$ | 0.5
$\epsilon_{Stack}$ | 1.1 × 10^3 J/kg K
$\Delta h_{f,liq}^\oplus$ | $-285.8$ kJ/mol

Table 8.5.: Parameter values for the stack cooling as used in the model

Figure 8.4.: Parameter Estimation of the Fuel Cell Characteristic for different values of the GDL thickness, left: limiting current used to express the concentration loss, right: concentration loss based on actual partial pressures at the reaction sites.

The graph on the left hand side of the figure 8.4 represents the estimations using the model which describes the concentration loss based on the limiting current whereas the graphs on the right hand side uses the model where the concentration loss is based on the actual reactant pressures at the reaction sites. The current limit of 261Ais shown in both plots. From the figure 8.4 it can be seen that the best estimation of the thickness of the GDL based on the experimental data is for the value of 209.75$\mu$m. Moreover, the
The corresponding electrochemical parameters are estimated by gPROMS,

\[
\begin{align*}
\zeta_1 &= -0.386739 \\
\zeta_2 &= 0.00129421 \\
\zeta_3 &= -4.04095 \times 10^{-5} \\
\zeta_4 &= 6.19985 \times 10^{-5} \\
\zeta_5 &= 0.00322244 \\
\zeta_6 &= -5.87561 \times 10^{-6} \\
\zeta_7 &= -2.58998 \times 10^{-6}
\end{align*}
\]

8.7.5. Results

The aim of this section is to develop a dynamic mathematical model of a polymer electrolyte membrane fuel cell based on first principles of chemistry, electrochemistry, and physics. The resulting model is capable of reflecting the fuel cell’s characteristic, although most of the models output cannot be validated through experimental data generally the results are as expected on a qualitative side. In the results is presented the whole operation cycle, which includes the start-up process, the normal operation and the shut-down, and also the transport of liquid and gases in the gas diffusion layer.

Complete Operation Cycle

In section 8.6 the operation procedure of the fuel cell stack is described (Figure 8.5). On the top right the stack load as input variable is displayed. The resulting power output of the stack, the volume of liquid water at the cathode reaction site - equivalent to the GDL section closest to the membrane - and the stack temperature are shown from the top right clockwise.

The Section I corresponds to the conditioning step and is followed by section II during which the flows and concentrations are allowed to stabilise. The Section III presents the results for the normal operation of the fuel cell stack for two load currents. During this operation period the stack experiences a step change in the load from 100\( \text{A} \) to 250\( \text{A} \). Finally the operation the fuel cell is shuts down as shown in section IV. The output variables has shown behaviour as was expected in qualitative terms. There is no power output during conditioning and shut-down, while during the normal operation there is a power output. When setting the load to the stack and the step change in the current there is a power overshoot before it settles at a constant value.
Figure 8.5.: Simulation of a complete operation cycle; top left: stack load as input variable, from top right clockwise: power output, liquid volume at the cathode reaction site and stack temperature as outputs.
Moreover there is no liquid water at the cathode reaction site before the operation period starts. Once the actual stack operation and therewith the reaction has started liquid water is immediately present. At the beginning there is a step change in the volume of water present followed by a very short period in which it is still increasing sharply and slightly overshooting the maximum value. With the step change in the current, there is a second increase in the amount of liquid volume present at the reaction site at the cathode. Finally, during the shut down process there is an obvious jump towards less water followed by a nearly linear decrease.

**Transport of Liquid Water**

The liquid water is transported from the membrane to the cathode channel and from there it is removed from the fuel cell as by-product. As can be seen from the Figure 8.6, before the operation starts there is no liquid water present in the gas diffusion layer at the cathode side. As soon as the electrochemical reaction starts there is liquid water formed and the amount present at the reaction site increases. Once a certain amount of water has accumulated it proceeds with its way through the membrane. After only around 40 sec of operation the water front reaches the channel. A steady state is reached with a distinct gradient across the gas diffusion layer. The amount of water remains highest where it is produced (intersection between gas diffusion layer and membrane).

**Gaseous Transport**

The concentration gradient that develops across the gas diffusion layer is representatively shown for oxygen in figure 8.7 for two different stack currents. The situations shown correspond to the conditions in the operation cycle introduced earlier at 1260 sec and 1850 sec. It represents the situation when the distribution of liquid water in the gas diffusion layer at the cathode side has reached equilibrium after a change in the stack current.

The two curves are nearly linear across the gas diffusion layer with the lower concentration values at the interface towards the membrane. As expected the gradient is much steeper when the current is higher. The channel concentration for the higher stack current is only slightly lower than for the
Figure 8.6.: Simulated transport of liquid water within the gas diffusion layer on the cathode side

Figure 8.7.: Simulated molar oxygen concentration within the gas diffusion layer on the cathode side
smaller current value.

8.8. Concluding Remarks

In this chapter we developed a dynamic mathematical model for a PEM fuel cell unit of 1kW. The model includes mass and energy models for the anode and cathode side and a distribution domain for the gas diffusion layer. The model validated by the experimental results of the polarization curve and managed to predict the gas diffusion layer thickness. Moreover, we present the analytical experimental design of this PEM fuel cell unit, which includes the electric load, transmitters, controllers, valves, pumps, thermocouples and the designed interface box. Finally, the safety and operational procedures of the unit are presented in order to satisfy the manufacturer specifications.
9. Modelling, Optimization and Control of a Tunnel Kiln

In this chapter we present a mathematical model for the firing process of a tunnel kiln. The dynamic mathematical model was validated through process data of real kiln plant for the entire firing zone. The validated model used to optimise the heat utilization by minimizing the total fuel consumption in the burners while maintain the quality specifications. The results showed that we manage to reduce the total fuel consumption by 8%.

9.1. Introduction

Tunnel kilns are widely used in the ceramics and brick making process industries (Kaya et al., 2009). Recently, the demand of bricks has dramatically increased in the economic growth countries due to infrastructure procedure. Although the brick-making industries in these countries are rapidly expanding their production, they are not optimally operated mainly due to the difficulty to collect data from the process in the plant which in return makes it difficult to identify the main parameters that affect the operation (ambient temperature, heat losses, complete oxidation, etc). In addition, the main factors that affect the price of the brick is the quality of the product, the fuel consumption, the production rate and the heat losses through the walls of the kiln (Kaya et al., 2008). Fuel consumption in the firing zone of a tunnel kiln comprises 80-85% of the total fuel consumed in brick production line, thus the production cost can be essentially reduced by improving the heat utilization (Kaya et al., 2009). The average energy requirement was reported to be approximately between 2040 and 3510kJ/kg of brick by Prasertsan Prasertsan et al. (1997). The industry mainly uses natural gas, Admixed Coal (AC) or Pulverized Coal (PC) as source of energy (Kaya et al., 2008).
A kiln is typically 40 to 150 m in length depending on the production capacity and operates with maximum temperature around 1000 °C. In the kiln the bricks go through three main processing zones, the preheating, firing and cooling - the operation of the kiln can then be characterised as a semi-batch process due to the burners operation in the firing zone (Figure 9.1). The air is supplied into the furnace at several locations (cooling zone and preheating) to satisfy the optimal temperature profile. The bricks enter the preheating zone with moisture content of 12% after the dryer. In the preheating zone, bricks are heated to evaporate the remaining water in order to avoid cracking due to thermal shock. The bricks reach the temperature of 700°C at the end of the preheating zone and their moisture content is reduced to less than 1%. During the firing zone the temperature is gradually increased to 1000°C by the combustion of natural gas on the top of the kiln (Figure 9.2). The produced heat is transferred to the bricks by convection and radiation while heat losses affect the temperature of the gas. Finally the bricks cool down to ambient temperature in the cooling zone, which is divided into fast cooling and slow cooling area. Blowing of the ambient air into the kiln at different locations along the cooling zone rapidly reduce the temperature of the bricks while increases the pressure drop.

There are many open issues for improving the energy efficiency of the tunnel kiln, which are mainly associated with the optimisation and control of the kiln process. For example manipulating the firing rate in the heating zone, the speed of the bricks in the tunnel and the air flowrate in the cooling zone, the energy consumption can be improved further while maintain the brick quality. In addition with the use of advanced control methods features, such as, constraints on temperature gradient to avoid cracking and on product quality can also be incorporated. These objectives can be achieved by developing a rigorous dynamic model of the process and then using it in an advanced model-based control and optimisation framework to improve the energy efficiency.

Despite the vast scope for improvements in the design, control and optimisation of tunnel kiln processes the techniques available to address these issues are rather limited. This can be attributed to the complexity of the process, arising from the nonlinear and dynamic nature of the model, from a computational point of view, and the lack of extensive and accurate measurements from the real plant. State-of-the-art model and optimisation
solution techniques have not yet been used to address these problems while the lack of real measurement data further obviates the need of developing a rigorous model and using state-of-the-art solution techniques to obtain reliable results.

Mathematical modelling of the tunnel kilns can be used as a tool to improve the operation conditions and design the appropriate control strategy in order to satisfy the quality constraints and minimize the fuel consumption. Another issue for the optimisation of the kiln operation is the optimal geometric configuration of the burners in the furnace and of the cooling fans. Limited studies have been appeared in the open literature to address the above issues. Dugwell and Oakley (1988) developed three different models to simulate one, two and three dimensional gas temperature profiles along the tunnel kiln. Their mathematical model represents the kiln as a series of plug-flow regions, in which heat transfer to ware occurs, interspaced by well-stirred adiabatic regions, in which burners and air inleakages are introduced. The models estimated the gas temperature and composition in addition to the refractory ware temperature. Halasz and Hangos (1988) proposed a one dimension distributed model in the form of integral equation. This model used to identify the optimal operation and stated that 5-8% energy saving was achieved. Yu (1994) presented a similar mathematical model describing the dynamic behaviour of the kiln. This model was used to predict the heat losses, gas flow and combustion of the fuels.

In 2005, Mancuhan (2006b) analysed the operation of a tunnel kiln producing bricks with a low or high calorific value coal added as an energy
They considered two different fuels; pulverized coal (PC) and natural gas (NG), and based their optimisation on a particular plant. The objective was to minimize the cost of the fuels used and the energy lost through the stack by solving a simplified 1D mathematical model. Following this work, Mancuhan (2006a) demonstrated how the optimal operating conditions could be predicted by using a mathematical model representing in the simplest form the phenomena of heat transfer, combustion of AC and PC, together with gas flow. The work focused on the firing zone of a tunnel kiln and used the same 1D model like in the previous work; they conclude that the heating values of AC and PC are the key parameters to obtain minimum fuel cost. Only one study has appeared in the open literature addressing the control problem of the furnace by P. Michael (2005). The controller design was based on fuzzy logic supervisory control and its aim was to improve the product quality while satisfy the safety limitations.

In this work a detailed mathematical model based on a single column design of bricks for the firing zone of the kiln has been developed. The one dimension model assumes uniform temperature and used to estimate the optimal operation of the kiln. The model comprises equations for fuel com-

Figure 9.2.: Average gas temperature
bustion, mass and energy balance, radiative and convective heat transfer and pressure drop. The resulting model involves algebraic equations and is solved by state-of-the-art numerical solution techniques by using gPROMS software (Limited), 2011). The model was simulated for various firing rates and results for heat loss, pressure drop, tile and car temperature profiles were obtained. One of the key achievements of this work is that despite the increased system complexity, the model validated through experimental data collected by an existing plant, and showed satisfactory approximation of the real process. Based on the validated model, we formulated and solved an optimisation problem with objective to minimize the fuel consumption and consequently the operational cost of the process. The modular nature of the model developed in this work allows it to be extended to different kiln designs and bricks allocation of the column. Finally a PI controller is designed for the entire firing zone to control the gas temperature by using as manipulated variables the secondary air and fuel mass flowrates.
### Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Area (m)</td>
<td></td>
</tr>
<tr>
<td>Cp</td>
<td>Heat capacity</td>
<td></td>
</tr>
<tr>
<td>Dh</td>
<td>Hydraulic diameter (m)</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Diffusion coefficient</td>
<td>x</td>
</tr>
<tr>
<td>e</td>
<td>roughness</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>Mass flowrate (kg/s)</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>Friction factor</td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>Convective heat coefficient (W/m°C)</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>Enthalpy</td>
<td></td>
</tr>
<tr>
<td>k</td>
<td>Thermal conductivity (W/mK)</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>Distance (m)</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Mass (kg)</td>
<td></td>
</tr>
<tr>
<td>MW</td>
<td>Molecular weight</td>
<td></td>
</tr>
<tr>
<td>Nu</td>
<td>Nusselt number</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Pressure (Pa)</td>
<td>b</td>
</tr>
<tr>
<td>ρ</td>
<td>Density (kg/m³)</td>
<td></td>
</tr>
<tr>
<td>Per</td>
<td>Perimeter, (m)</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>Heat transfer (kJ)</td>
<td>sec</td>
</tr>
<tr>
<td>r</td>
<td>Reaction rate</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Ideal gas law (J/kg K)</td>
<td>i</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Greek Letters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>Friction factor</td>
</tr>
<tr>
<td>ε</td>
<td>Emissivity (-)</td>
</tr>
<tr>
<td>h</td>
<td>Convective heat coefficient (W/m°C)</td>
</tr>
<tr>
<td>ΔHr</td>
<td>Reaction enthalpy</td>
</tr>
<tr>
<td>ε</td>
<td>Emissivity (-)</td>
</tr>
<tr>
<td>∆P</td>
<td>Pressure Drop (Pa)</td>
</tr>
<tr>
<td>σ</td>
<td>Stefan-Boltzman coefficient (W/m² K⁴)</td>
</tr>
<tr>
<td>µ</td>
<td>Viscosity (kg/m s)</td>
</tr>
</tbody>
</table>

### Subscripts

<table>
<thead>
<tr>
<th>Description</th>
<th>Subscript</th>
</tr>
</thead>
<tbody>
<tr>
<td>Air</td>
<td>air</td>
</tr>
<tr>
<td>Brick</td>
<td>b</td>
</tr>
<tr>
<td>Car</td>
<td>car</td>
</tr>
<tr>
<td>Fuel</td>
<td>fuel</td>
</tr>
<tr>
<td>Per</td>
<td>Per</td>
</tr>
<tr>
<td>Free</td>
<td>Free</td>
</tr>
<tr>
<td>Inlet</td>
<td>in</td>
</tr>
<tr>
<td>Component i</td>
<td>i</td>
</tr>
</tbody>
</table>

## 9.2. Mathematical Model

Furnace operation is based on the counter current flow of the bricks and gas which means that the bricks are moving from the preheating to the cooling zone while gas is moving in opposite direction. The gas is a mixture of nitrogen, oxygen, carbon dioxide and vapour produced by the combustion of natural gas. The tunnel kiln is composed into three zones, preheating, firing and cooling as shown in Figure 9.1. The operation of the kiln is characterised as semi-batch due to the combustion in the firing zone. The burners operate for 250 seconds continuously and then they stop in order
the car to move along the kiln. The design of the tunnel kiln is based on a real plant operating in Southern Europe with the following geometry of the furnace and gas temperature range (Figure 9.2),

1. Preheating zone is 50m and temperature range is 25-700°C
2. Firing zone is 27m and temperature range is 700-1000°C
3. Cooling zone is 42m and temperature range is 900-40°C

The proposed one dimensional dynamic mathematical model considers each column of the car individually and assumes continuous operation for simplification purposes. Figure 9.3) explains the proposed column layout based on which the mathematical model has been designed. The mathematical model consists of mass balances, energy balances and semi-empirical equations for the pressure drop estimation. The main assumptions are the following

1. Dynamic operation
2. The gas has uniform temperature and velocity
3. The variation of the temperature and flowrates in the vertical direction was not taken into account for simplification.
4. Complete and spontaneous oxidation of the natural gas on the roof of the kiln
5. Ideal gas law holds in the gas phase
6. The heat is transferred between the gas and the bricks by convection and radiation
7. The bricks and gas have uniform temperature at each column
8. The bricks properties are function of the temperature
9. The heat losses are through the roof and walls of the kiln and depend on the ambient temperature
10. The car moves along the tunnel with a constant velocity
9.2.1. Mass balances

The mass balances consist of the main gas stream which is coming from the previous column, the fuel and the secondary air which is introduced from the roof (Figure 9.2). The secondary air is required for the complete oxidation of the natural gas and to reduce the percentage of the carbon dioxide in the furnace. However, it is ideal to keep low because it has cooling effects in the firing zone, while is ideal to be used in the preheating zone.

The reaction takes place at the roof of the kiln, where the burners are placed and is instantly. The following equations describe the mass balances for the gas in the firing zone. The conservation law to the mass of component i in the system is

\[
\frac{dM_i}{dt} = F_{\text{air}}x_{\text{air},i} + F_{\text{fuel}}x_{\text{fuel},i} + F_{\text{sec}}x_{\text{sec},i} - F_{\text{out}}x_{\text{out},i} + r_i \quad (9.1)
\]

The total holdup MT is related to the component holdups, and depends on the density of the gas in the system.

\[
M_T = \sum_{i=1}^{5} M(i) \quad (9.2)
\]
\[ M_T = V_{free} \rho_{out} \]  
(9.3)

\[ \rho_{out} = f(T_{out}, P_{out}, x_{out,i}) \]  
(9.4)

The composition can be expressed in the terms of the mass fraction \( x_i \), \( i = 1, \ldots, 5 \)

\[ M_i = M_T x_{out,i} \]  
(9.5)

Finally, the outlet flowrate \( F_{out} \) is given by the following equations and depends on the pressure, density and velocity

\[ F_{gas\_out} = A_{free} \rho_{gas} u \]  
(9.6)

\[ F_{out} = f(P_{in}, P_{out}) \]  
(9.7)

**Semi-empirical equations (heat transfer coefficient & pressure drop)**

The heat transfer coefficient of the gas is a function of the temperature, pressure and velocity of the gas through the column and is not constant along the kiln. The calculation is based on semi-empirical equation based on the Nusselt, Reynolds and Prandtl numbers (O.B. Gol’tsova, 2006).

\[ Nu = \frac{h D_h}{k} \]  
(9.8)

\[ Nu = 0.021 Re^{0.8} Pr^{1/3} \]  
(9.9)

\[ Re = \frac{u D_h \rho_{out}}{\mu_{out}} \]  
(9.10)

\[ Pr = \frac{C_{p_{out}} \mu_{out}}{k_{out}} \]  
(9.11)

The hydraulic diameter depends on the bricks arrangement in the column and usually is between 1 and 2 (Yu, 1994)

\[ D_h = \frac{4 A_b}{Per} \]  
(9.12)
In the furnace the gas flows through the space between the bricks, thus there is a significant pressure drop depending on the bricks arrangement. The pressure drop can be calculated based on the Fanning friction factor, $f$, by the following equations

$$\Delta P = P_{in} - P_{out} \quad (9.13)$$

$$\delta P = \frac{4LJp_{out}u^2}{2D_h} \quad (9.14)$$

The friction factor depends on the flow regime and for turbulent flow can be calculated from the Colebrook correlation (Kaya et al., 2008)

$$\frac{1}{\sqrt{f}} = -2\log \left( \frac{2.51}{Re\sqrt{f}} + \frac{e}{D_h} \right) \quad (9.15)$$

We assume that the flow is turbulent during the process, thus we are not considering the equation for laminar flow.

### 9.2.2. Energy Balances

The energy conservation for the gas can be written as follows

$$\frac{dU_{out}}{dt} + V_{free} \frac{dP_{out}}{dx} = H_{in} - H_{out} - Q_{car} - Q_{tiles} - Q_{loss} + \Delta H_r \quad (9.16)$$

Under the assumption that the pressure doesn’t change significantly in each column, the term $dP_{out}$ becomes zero. The last term in the above equation is the heat released by the combustion of the natural gas.

The total enthalpy holdup is related to the specific enthalpy

$$H_T = M_T h \quad (9.17)$$

This equation determines the specific system enthalpy which varies with time and depends on the temperature, pressure and composition of the gas.

$$h = f(T, P, x) \quad (9.18)$$

The expressions for the heat exchange between the gas and car and bricks are as follows (Yu, 1994)
The temperature of the car is increasing in the firing zone due to the convection between the car and the gas.

\[ Q_{\text{car}} = h_{\text{car}} A_{\text{car}} \left( T_{\text{out}} - T_{\text{car, out}} \right) \]  
\[ Q_{\text{b}} = h A_{\text{brick}} \left( T_{\text{out}} - T_{\text{b}} \right) \]

The differential energy balance for the bricks is given by

\[ M_{\text{b}} c_p \frac{dT_{\text{b}}}{dt} = H_{\text{b,in}} - H_{\text{b, out}} + Q_{\text{b}} + Q_{\text{rad}} \]  

The last term is the radiation heat produced by the reaction and affects only the bricks and wall (Kaya et al., 2009).

\[ Q_{\text{rad}} = {\sigma F_b \varepsilon \alpha_b(T_{\text{out}}^4 - T_{\text{b}}^4) \]  

The heat losses are consisting of the heat removing the kiln by the wall and roof by convection and radiation between the kiln walls and gas.

\[ Q_{\text{losses}} = Q_w + Q_r \]

\[ Q_w = A_w h_w (T_{\text{out}} - T_{\text{w}}) + Q_{\text{rad}, w} \]

\[ Q_r = A_r h_r (T_{\text{out}} - T_{\text{r}}) + Q_{\text{rad}, r} \]

Finally we estimate the energy removing the kiln through the walls and the roof. Heat transfer through walls is due to the temperature difference between the two faces of a wall. It depends on the heat resistance of the wall material \( k_{\text{kiln}} \), the ambient temperature \( T_{\text{amb}} \) and the thickness \( (L_w, L_r) \)

\[ k_{\text{kiln}} A_w \frac{T_w - T_{\text{amb}}}{L_w} = Q_w \]

\[ k_{\text{kiln}} A_r \frac{T_{\text{roof}} - T_{\text{amb}}}{L_r} = Q_r \]
The total model for the firing process includes 3348 variables and 216 differential equations and solved with gPROMS in 19 seconds for simulation period 4,000 seconds. This simplified model was developed to estimate the behaviour of the process and optimise the performance of the furnace by reducing the fuel consumption.

9.2.3. Model Validation

The derived dynamic mathematical model is validated through process data from an existing tunnel kiln. The tunnel kiln has 54 series of burners on the roof and the total average fuel consumption is 0.012kg/kg brick. The used data are the following:

1. Mass flowrate of the main air stream
2. Pressure in the kiln
3. Gas temperature inside the furnace
4. Total fuel consumption
5. Dimension of the tunnel, the bricks and the car
6. Total load of each column

All model simulations and parameter estimation experiments were implemented in the advanced process modelling environment gPROMS. gPROMS is an equation-oriented modelling system used for building, validating and executing first-principles models within a flowsheeting framework. Parameter Estimation in gPROMS is based on the Maximum Likelihood formulation which provides simultaneous estimation of parameters in both the physical model of the process as well as the variance model of the measuring instruments. gPROMS attempts to determine values for the uncertain physical and variance model parameters, that maximise the probability that the mathematical model will predict the measurement values obtained from the experiments. The formulation of the problem leads to the following recursive least squares parameter estimation if no variance model is selected.
Table 9.1.: Temperature results for the validation of the firing process

<table>
<thead>
<tr>
<th>Distance (m)</th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
<th>24</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{\text{process}}$ ($^\circ$C)</td>
<td>586</td>
<td>805</td>
<td>920</td>
<td>950</td>
<td>976</td>
<td>972</td>
<td>937</td>
<td>898</td>
</tr>
<tr>
<td>$T_{\text{gas}}$ ($^\circ$C)</td>
<td>660</td>
<td>832</td>
<td>903</td>
<td>944</td>
<td>930</td>
<td>942</td>
<td>920</td>
<td>910</td>
</tr>
<tr>
<td>$T_{\text{brick}}$ ($^\circ$C)</td>
<td>641</td>
<td>802</td>
<td>876</td>
<td>934</td>
<td>910</td>
<td>930</td>
<td>945</td>
<td>930</td>
</tr>
</tbody>
</table>

\[
\min_{F_i,T_s} \sum_{i=1}^{N} (T_{\text{process},i} - T_{\text{gas},i})^2 + \sum_{i=1}^{N} (F_{f,i} - F_{f,\text{process},i})^2
\]

s.t.

\[eq.9.1 - eq.9.28\]

(9.29)

where $T_{\text{process},i}$, $T_{\text{gas},i}$ is the process and model temperature for the column $i$ and $F_i$ is the mass flowrates of all the streams, $F_{f,i}$ and $F_{f,\text{process}}$ is the mass flowrate of the fuel for the simulation and the actual process, respectively.

Figure 9.4 presents the results of the simulation and the actual process and shows that the developed model manages to approximate the behaviour of the process even though there is an error, probably due to the missing information for the heat transfer coefficients. Finally, the total fuel consumption is 0.014kg/kg brick (simulation) which is close to the actual process (0.012kg/kg brick). Better estimation could be obtained if there were additional information for the fuel consumption for each burner individually, the exact kiln heat transfer coefficients, as well as the exact temperature of the secondary air.

The derived dynamic mathematical model is validated through process data from an existing tunnel kiln. The tunnel kiln has 54 series of burners on the roof and the total average fuel consumption is 0.012kg/kg brick. The used data are the following:

1. Mass flowrate of the main air stream
2. Pressure in the kiln
3. Gas temperature inside the furnace
Figure 9.4 presents the results of the simulation and the actual process and shows that the developed model manages to approximate the behaviour of the process even though there is an error, probably due to the missing information for the heat transfer coefficients. Finally, the total fuel consumption is 0.014 kg/kg brick (simulation) which is close to the actual process (0.012 kg/kg brick). Better estimation could be obtained if there were additional information for the fuel consumption for each burner individually, the exact kiln heat transfer coefficients, as well as the exact temperature of the secondary air.

In our process we assumed that the temperature of the secondary air was constant and around 450°C since it comes from the cooling process.

<table>
<thead>
<tr>
<th>Inlet air mass flux (kg/m²s)</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlet air mass flux (kg/m²s)</td>
<td>1.4</td>
</tr>
<tr>
<td>Total fuel consumption (kg/kg brick)</td>
<td>0.014</td>
</tr>
<tr>
<td>Total secondary air (kg/m²s)</td>
<td>0.3</td>
</tr>
<tr>
<td>Secondary air temperature (°C)</td>
<td>450</td>
</tr>
<tr>
<td>Outlet CO₂ mass fraction</td>
<td>3%</td>
</tr>
</tbody>
</table>

Figure 9.4.: Process and simulation gas temperature

Table 9.2.: Massflowrate results for the validation of the firing process
Moreover, it was observed that the ambient temperature is a critical factor (disturbance) for this type of systems and should take into consideration in the design of the controller. Finally, we have to mention here that we have no information for the temperature of the bricks since all the sensors are placed to measure the gas temperature. The results of the model validation are presented in Figures 9.4, 9.5 and in Tables 9.1 and 9.2, where the main flowrate and temperatures for the simulation model are presented. The main benefit of the model is that it estimates the temperature of the bricks across the firing zone based on first principle mathematical modelling.

9.2.4. Controller design

A proportional-integral (PI) controller is initially designed in the optimization part for manipulating the fuel and secondary air flowrates to reach the optimal temperature profile and consequently to meet the quality specifications. The controller gains, $K_C$ and $\tau_I$, were tuned to minimize the ISE incurred up to the furnace reach the temperature profile. The manipulated variables of the system is the mass flowrates of the fuel and the secondary air, $U_f$, and the controlled variable is the gas temperature in each column ($T_{gas}$), $KC$ the controller gain, $\tau_i$ the integral time and $e = T_{gas} - T_{sp}$ the error between the reference temperature profile, and the gas temperature of the kiln.
The gains of the controller are obtained by minimizing the Integral Square Error criterion and are given by $K_C = 0.01$ and $\tau_i = 10$ sec.

Simulation results for PI controller are presented in Figure 9.6 and shows that the designed controller managed to reach the set point temperature profile along the firing zone with a small offset at the beginning in the first four columns. Furthermore, we present the results for the bricks temperature comparing with the gas temperature in Figure 9.7.

### 9.2.5. Optimal Kiln Operation

The main objective in the tunnel kilns is to improve the heat efficiency while maintain the temperature profile in order to satisfy the quality specification. Thus, we formulated and solved an optimisation problem to investigate the minimum fuel consumption in the firing zone of the furnace. The problem formulation is the following
Figure 9.7.: Simulation brick temperature for the PI controller

\[
\min_{F_{f,j}} \sum_{i=1}^{i=N_j} F_{f,j} \\
\text{s.t. } \mu \leq T_{\text{gas},j} - T_{\text{sp},j} \leq M \\
C_{\text{CO}_2,j} \leq \Delta C_{\text{max}} \\
T_{b,j} - T_{\text{gas},j} \leq \Delta T_{\text{max}} \\
eq.9.1 - \eq.9.28
\]  

(9.32)

where \( F_{f,j} \) is total mass flowrate of the natural gas for the \( j \) column, \( C_{\text{CO}_2} \) is the mass concentration of the carbon oxide, \( M \) is the upper bound temperature error from the set point, \( \mu \) is the lower bound error, \( T_{\text{gas}} \) and \( T_{\text{sp}} \) is the temperature of the gas and temperature set point in column \( j \), respectively.

The process temperature (given by the available data) is used as a set-point in the above calculations, although they were different from the actual gas temperature inside the kiln.

The optimisation variables in the firing zone are

1. The mass flowrate of the fuel in each column, \( F_{f,j} \)
2. The mass flowrate of the secondary air in each column, \( F_{s,j} \)
3. The mass flowrate of the air coming from the cooling zone, \( F_{c} \)
The derived dynamic nonlinear optimisation problem consists of 109 optimisation variables, 2258 continuous equations and 216 initial conditions. The continuous problem formulated and solved in gPROMS in 653 sec, and the results for the tiles and gas temperature are presented in Figure 9.8 and table 9.3. The method we used for solving problem (9.32) is the multi-shooting algorithm for dynamic optimization (as implemented in gPROMS with the name CVP_MS). During the optimisation procedure, the optimiser obtains the duration of each control interval, the values of the control variables over that interval, and additionally, the values of the differential variables at the start of each control interval. This information is used to determine the values of the objective function and constraints. Finally, based on this solution, the optimiser revises the choices it made at the first step, and repeats the above procedure until it obtains a feasible point with minimum objective function.

The optimisation results are presented in Tables 9.3, 9.4 and Figures 9.8 and 9.9. As we can see from table 9.4 the fuel consumption is reduced by 8%, without changing the temperature of the tiles significantly thus maintain the tile quality. Mainly, the secondary air flow has been reduced as well as the mass flowrate of the main air stream coming from the cooling zone.
Table 9.3.: Temperature results for the optimal firing process

<table>
<thead>
<tr>
<th>Distance (m)</th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>12</th>
<th>16</th>
<th>20</th>
<th>24</th>
<th>28</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{process} (°C)</td>
<td>586</td>
<td>805</td>
<td>920</td>
<td>950</td>
<td>976</td>
<td>972</td>
<td>937</td>
<td>898</td>
</tr>
<tr>
<td>T_{gas} (°C)</td>
<td>750</td>
<td>906</td>
<td>893</td>
<td>916</td>
<td>942</td>
<td>938</td>
<td>935</td>
<td>811</td>
</tr>
<tr>
<td>T_{brick} (°C)</td>
<td>612</td>
<td>838</td>
<td>928</td>
<td>943</td>
<td>919</td>
<td>902</td>
<td>902</td>
<td>880</td>
</tr>
</tbody>
</table>

Table 9.4.: Massflowrate results for the optimal firing process

<table>
<thead>
<tr>
<th>Massflowrate parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet air mass flux (kg/m²s)</td>
<td>0.85</td>
</tr>
<tr>
<td>Outlet air mass flux (kg/m²s)</td>
<td>1.17</td>
</tr>
<tr>
<td>Total fuel consumption (kg/kg brick)</td>
<td>0.013</td>
</tr>
<tr>
<td>Total secondary air (kg/m²s)</td>
<td>0.2</td>
</tr>
<tr>
<td>Secondary air temperature (°C)</td>
<td>600</td>
</tr>
<tr>
<td>Outlet CO₂ mass fraction</td>
<td>2.9%</td>
</tr>
</tbody>
</table>
9.2.6. Concluding Remarks

This work was focused with a systematic study of the modelling and optimization and control of a tunnel kiln where the main objective is the energy management and savings of the process. We presented a dynamic mathematical model for the design of a column in the firing zone based on first principles. The developed mathematical model was validated, by using real process data and it was shown that the gas temperature prediction of the model matched the temperature across the firing zone. Finally, a model based dynamic optimization was performed by using multi-shooting algorithm to improve energy utilization in the firing zone and managed to reduce the natural gas consumption in the simulation model by 8%. The developed model is generic and can be extended to different kiln design and bricks allocation of the column. Moreover a PI controller has been designed with satisfactory performance for the gas temperature control. The obtained results also provide the confidence for carrying out model-based control design in order to achieve better performance than the presented and also to include in the controller design operational constraints and known disturbances such as the ambient temperature.
10. Conclusions and Future Research Directions

This chapter is a summary of the major findings and contributions of the work described in this thesis. Directions for future developments based on this work are also identified.

10.1. Conclusions

Part I has focused on the development of explicit/multi-parametric controllers for linear models based on the dynamic programming. In chapter 2 is presented a new algorithm for solving the explicit/multi-parametric Model Predictive Control (MPC) problem for linear, time-invariant discrete-time systems, based on dynamic programming and multi-parametric programming techniques. The algorithm features two key steps, i) a dynamic programming step, in which the mp–MPC problem is decomposed into a set of smaller sub–problems where only the current control, state variables and constraints are considered, and ii) a multi–parametric programming step in which each sub–problem is solved as a convex multi–parametric programming problem, to derive the control variables as an explicit function of the states. The key feature of the proposed method, is that it overcomes potential limitations of previous methods for solving multi–parametric programming problems with dynamic programming, such as the need for global optimization for each sub–problem of the dynamic programming step. Extensions of the proposed procedure for the case of hybrid systems (corresponding to mixed–integer programming problems), continuous–time systems (where the mp–MPC optimization becomes a multi–parametric Dynamic Optimization problem) and non–linear system (involving multi–parametric nonlinear programming methods) pose significant challenges, due to the more complex optimization problems involved in mp–MPC.
In chapter 3 a new algorithm for robust explicit/multi-parametric Model Predictive Control (MPC) for uncertain, linear discrete-time systems is proposed, based on the work of chapter 2. The developed algorithm features, i) a DP reformulations of the MPC optimization problem, ii) a robust reformulation of the constraints, and iii) a multi-parametric programming step, where the control variables are obtained as explicit functions of the state variable, such that the state and input constraints are satisfied for all admissible values of the uncertainty. A key feature of the proposed procedure, is that, as opposed to previous methods, it only solves a convex multi-parametric programming problem for each stage of the DP procedure.

In chapter 4 has been developed an algorithm for Robust explicit/multi-parametric controllers, which are designed for constrained, linear discrete-time systems with box-constrains on states and inputs variables and involving uncertainty in the left-hand side (LHS) of the linear model. The main features of the algorithm are: (i) a dynamic programming reformulation of the MPC optimization, (ii) a robust reformulation of the constraints that accounts for uncertainty and (iii) a multi-parametric programming solution step where the controls are obtained as an explicit function of the states.

Part I has focused on the dynamic mathematical modelling of complex energy processes, such as metal hydride, PEM fuel cell stack and unit and tunnel kiln for a brick plant. In addition, the analytic design of a PEM fuel cell unit has been presented which is developing for the new lab of Imperial College. Chapter 5, presents a detailed two-dimension dynamic process model for a hydrogen storage tank. The developed model has been optimized to identify the optimal operation of the process. Finally an explicit/multi-parametric model predictive controller has been designed. For the controller design, a reduced order approximate model is obtained, based on which nominal and robust multi-parametric controllers are presented.

In chapter 6, an analytical dynamic model for a PEM fuel cell unit is presented. The described model consist of of a detailed model for the PEM fuel cell stack and simplified models for the compressor, humidifier and cooling system. This process model has been used to derive a reduced order linear model for the design of an explicit controller. The designed controller is tested and validated on several operating condition.

In chapter 7 two cases of controller design for PEM fuel cell units is pre-
The first case describes the controller design of a fully automated integrated Hydrogen Fuel Cell Testing Unit (HFCTU) at CERTH. The controller was based on the reduced order model developed by simulation results of the dynamic mathematical model. The controller was able to satisfy the load demand, avoid starvation and maintain the temperature at a nominal point. The second case involves the temperature and oxygen excess ratio controller of a 1.2 kW PEM fuel cell system by Ballard (Nexa Power Module) at University of Seville. The objective of this controller design was to maintain the stack temperature and the oxygen excess ratio at several set-points in order to test the performance of the derived controller.

In chapter 8 is described the analytical design of an experimental unit for a 1kW PEM fuel cell by NedStack. The unit consists of the electric load, two humidifiers, cooling/heating system, valves, filters, and temperature, flow and pressure sensors. The unit is in the development phase and it is expected to be fully operational in 2-3 months period. Moreover, the mathematical model of the PEM fuel cell stack is developed and validated based on the data received by the manufacturer. Finally, in Appendix III is developed a dynamic mathematical model for the firing process of a tunnel kiln based on . The dynamic mathematical model was validated through process data of real kiln plant for the entire firing zone. The validated model used to optimise the heat utilization by minimizing the total fuel consumption in the burners while maintain the quality specifications. The results showed that we manage to reduce the total fuel consumption by 8%. Additionally, a PI controller was designed to control the gas temperature in the firing zone.

10.2. Key contributions of thesis

The key contributions of this thesis can be summarised as follows:

- a novel explicit/multi-parametric Model Predictive Controller (MPC) design for time-invariant discrete-time systems, based on dynamic programming;

- a new explicit Robust Model Predictive Controller (MPC) design for uncertain time-invariant discrete-time systems, based on dynamic programming;
• an algorithm for Robust explicit/multi-parametric controllers, which are designed for constrained, linear discrete–time systems with box-constrains;

• a two-dimension dynamic mathematical model for a metal hydride storage tank reactor and the controller design of the unit;

• the dynamic mathematical model of PEM fuel cell unit;

• the analytical design of an experimental PEM fuel cell unit;

• a new dynamic mathematical model for the firing process of a tunnel kiln;

10.3. Future Work

In this section, suggestions for future work are discussed.

10.3.1. Theoretical and algorithm developments

Multi-parametric programming has been used widely the last 10 years in the area of control and more precisely in the area of MPC, since it is able to generate the optimal maps(solutions) for a set of parameters. However the main limitation of all the existing algorithms are the computation complexity, which increases as the constraints and horizon increases. The algorithm presented in chapter 2 can be a good alternative if there is an alternative way to merge the solution at each step with an error criterion able to guarantee the feasibility and optimality of the solution.

In chapter 3 and 4 has been presented an algorithm for robust explicit MPC of a linear state space system with uncertainty on the system matrices. Possible developments in this promising algorithm are listed as follows

• This methodology can guarantee stability and feasibility of the linear system, however it generates many optimal solutions(regions). An algorithm who will be able to merge the solutions at each step with an additional criterion error which will be able to guarantee the existence of feasible merge solution will increase the computation time and will allow the algorithm to implement in larger problems.
The methodology will be further improved if the objective function instead of solving a minimisation problem alter to solve a min-max problem with incorporation of the uncertainty in the objective function. This will allow the algorithm to guarantee also the optimality in all the possible set of uncertainties and will improve the robustness of the algorithm.

The algorithm for robust explicit MPC can be applied for the case of hybrid systems with additional modification in the objective function and by the use of mixed integer multi-parametric programming algorithm.

10.3.2. Multi-parametric Programming

The advantage of multiparametric programming and MPC is their ability to replace the online optimization with computationally inexpensive function evaluations, that can be stored in hardware. This paves the way for many advanced control applications not only in the area of large and medium scale processes, where advanced control and MPC has been traditionally applied, but also for small-scale systems such as portable devices and equipment, where advanced control methods had not yet found applications due to the insufficient computational power required for their implementation.

10.3.3. Explicit nonlinear MPC (mp–NMPC)

While initial and recent developments are quite promising, this is an area that is very much at early stages. Any developments here will of course depend on fundamental developments in the area of multi-parametric nonlinear programming (mp–NLP), multi-parametric dynamic optimization (mp–DO), nonlinear dynamic programming by multi-parametric programming and global optimization. Furthermore, developments are also expected for the case of nonlinear systems involving 0-1 variables, i.e. hybrid systems – here algorithmic developments for the solution of general mixed–integer nonlinear and dynamic optimization (mp–MILP, mp–MIDO) will be involved.
10.3.4. Robust multi-parametric/explicit control of hybrid and nonlinear systems

Recent developments have started to address the design of robust multi-parametric controllers for linear MPC systems, there is clearly a need to establish robust controller design methods for general classes of nonlinear systems, within a mp–NMPC framework. Furthermore, theory for the design of robust multi-parametric controllers of continuous-time, dynamic systems is almost completely lacking. Here, advances in robust optimization of dynamic systems and multi-parametric dynamic optimization (mp–DO) are needed. A further challenge arises when 0-1 binary variables are present in the model, i.e. for hybrid systems. For linear systems, robust optimization results for MILP models offer an excellent starting point; however, there is a lack of theory algorithms for the robust solution of general mixed-integer nonlinear and dynamic optimization.

10.3.5. Design linear feedback gain with additive disturbances

Recently, great attention has been given to the design of linear robust algorithms for the cases of additive disturbances, however the problems cannot be solved for infinite horizon due to complexity and computation issues. Usually, the robust controller is designed to track the states to the minimal robust invariant set and then optimal feedback gain is applied (Ricatti) which guarantees that the system will remain inside the mRPI set. In this approach we try to find the linear feedback gain $K$, for which 1) the mRPI set is the minimum possible (minimum distance of all the vertices from the origin) and 2) the linear $K$ which satisfy the constraints. In this section we suggest an preliminary approach to design a feasible linear feedback gain which could guarantee feasibility and disturbance rejection of the linear system. The described problem is based on the work of Rakovic et al. (2011) and leads to a bilinear problem. Two different formulations are presented, where the first is trying to find the min-max mRPI set (minimum distance of all the vertices from the origin) while the second aims to design a linear feedback gain which guarantees that will be feasible for the entire state space map. Clearly, the described methodology doesn’t find the optimal input sequence, but a linear feedback gain which satisfies the constraints for
a linear system with additive disturbances.

The basic nomenclature and definitions are the followings. The sets of non-negative and positive integers are denoted, respectively by $N := \{0, 1, 2, \ldots\}$ and $N_+ := \{1, 2, \ldots\}$. Let $N_{q_1,q_2} := \{q_1, q_1 + 1, \ldots, q_2\}$ for given $q_1, q_2 \in N$ such that $q_1 \leq q_2$. Further a set of non-negative real numbers is denoted by $\mathbb{R}_+$, so that $\mathbb{R}_+ := \{x \in \mathbb{R} : x \geq 0\}$. The Minkowski set addition of two non-empty subsets if $\mathbb{R}^n$, say $X$ and $Y$, is denoted by $X \oplus Y := \{x + y : x \in X, y \in Y\}$, $\text{int}(X)\text{denotes the interior of } X$. Given a sequence of sets $\{X_i \subset \mathbb{R}^n\}_{i=a}^b$, we denote $\bigoplus_{i=a}^b X_i = X_a \oplus X_{a+1} \oplus \ldots \oplus X_b$.

**Brief summary and general concept-Problem formulation**

Given a system

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (10.1)$$

State, control ad terminal constraint sets are given by

$$X := \{x : \forall i \in N_{[1,r]}, \ F_i^T x \leq 1\} \quad (10.2)$$

$$U := \{u : \forall i \in N_{[1,s]}, \ G_i^T u \leq 1\} \quad (10.3)$$

$$X_f := \{x : \forall i \in N_{[1,t]}, \ H_i^T x \leq 1\} \quad (10.4)$$

where $N_{[1,l]} : \{1, 2, \ldots, l\}$ for a positive integer $l$. The disturbance set is given as a convex hull of its known extreme points:

$$\mathcal{W} := \text{convh}(\{w_i : i \in N_{[1,q]}\}) \quad (10.5)$$
Main concept

<table>
<thead>
<tr>
<th>$X_{(0,N)}$</th>
<th>$X_{(0,0)}$</th>
<th>$X_{(0,1)}$</th>
<th>$X_{(0,2)}$</th>
<th>$\ldots$</th>
<th>$X_{(0,N-1)}$</th>
<th>$X_{(0,N)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_{(0,N-1)}$</td>
<td>$U_{(0,0)}$</td>
<td>$U_{(0,1)}$</td>
<td>$U_{(0,2)}$</td>
<td>$\ldots$</td>
<td>$U_{(0,N-1)}$</td>
<td></td>
</tr>
<tr>
<td>$X_{(1,N)}$</td>
<td>$X_{(1,1)}$</td>
<td>$X_{(1,2)}$</td>
<td>$\ldots$</td>
<td>$X_{(1,N-1)}$</td>
<td>$X_{(0,N)}$</td>
<td></td>
</tr>
<tr>
<td>$U_{(1,N-1)}$</td>
<td>$U_{(1,1)}$</td>
<td>$U_{(1,2)}$</td>
<td>$\ldots$</td>
<td>$U_{(1,N-1)}$</td>
<td>$\ldots$</td>
<td></td>
</tr>
<tr>
<td>$X_{(2,N)}$</td>
<td>$X_{(2,1)}$</td>
<td>$\ldots$</td>
<td>$X_{(2,N-1)}$</td>
<td>$X_{(0,N)}$</td>
<td>$\ldots$</td>
<td></td>
</tr>
<tr>
<td>$U_{(2,N-1)}$</td>
<td>$U_{(2,1)}$</td>
<td>$\ldots$</td>
<td>$U_{(2,N-1)}$</td>
<td>$\ldots$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$X_{(N-1,N)}$</td>
<td>$\ldots$</td>
<td>$X_{(0,N)}$</td>
<td></td>
<td></td>
<td>$\ldots$</td>
<td></td>
</tr>
<tr>
<td>$U_{(2,N-1)}$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td></td>
<td>$\ldots$</td>
<td></td>
</tr>
<tr>
<td>$X_{(N,N)}$</td>
<td></td>
<td>$\ldots$</td>
<td></td>
<td>$\ldots$</td>
<td>$\ldots$</td>
<td></td>
</tr>
<tr>
<td>$X_N$</td>
<td>$X_0 = X_{(0,0)}$</td>
<td>$X_1 = \bigoplus_{j=0}^1 X_{(j,1)}$</td>
<td>$X_2 = \bigoplus_{j=0}^2 X_{(j,2)}$</td>
<td>$\ldots$</td>
<td>$X_{N-1} = \bigoplus_{j=0}^{N-1} X_{(j,N-1)}$</td>
<td>$X_N = \bigoplus_{j=0}^N X_{(j,N)}$</td>
</tr>
<tr>
<td>$U_{N-1}$</td>
<td>$U_0 = U_{(0,0)}$</td>
<td>$U_1 = \bigoplus_{j=0}^1 U_{(j,1)}$</td>
<td>$U_2 = \bigoplus_{j=0}^2 U_{(j,2)}$</td>
<td>$\ldots$</td>
<td>$U_{N-1} = \bigoplus_{j=0}^{N-1} U_{(j,N-1)}$</td>
<td>$\ldots$</td>
</tr>
</tbody>
</table>

Problem Constraints-Main Decision Variables

Introduce:

$$x_0 := (x_{(0,0,0)}, x_{(0,0,1)}, \ldots, x_{(0,0,N)})^T \in \mathbb{R}^{(N+1)n} \quad (10.6)$$

$$\forall k \in \mathbb{N}_{[1:N]}, \forall i \in \mathbb{N}_{[1:q]}, \quad x(i,k) := (x(i,k,k), x(i,k,k+1), \ldots, x(i,k,N))^T \in \mathbb{R}^{(N+1-k)n}, \quad (10.7)$$

$$u_0 := (u_{(0,0,0)}, u_{(0,0,1)}, \ldots, u_{(0,0,N)})^T \in \mathbb{R}^{Nm} \quad (10.8)$$

$$\forall k \in \mathbb{N}_{[1:N]}, \forall i \in \mathbb{N}_{[1:q]}, \quad u(i,k) := (u(i,k,k), u(i,k,k+1), \ldots, u(i,k,N-1))^T \in \mathbb{R}^{(N-k)m}, \quad (10.9)$$

Furthermore introduce for all $k \in \mathbb{N}_{[1:N]}$

$$x_k := (x_{(1,k)}^T, x_{(2,k)}^T, \ldots, x_{(q,k)}^T)^T \in \mathbb{R}^{q(N+1-k)m} \quad (10.10)$$

233
\[ u_k := (u_{(1,k)}^T, u_{(2,k)}^T, \ldots, u_{(q,k)}^T)^T \in \mathbb{R}^{q(N-k)m} \tag{10.11} \]

\[ k_k := (k_{(1,1)}, k_{(1,2)}, \ldots, u_{(n,m)}^T)^T \in \mathbb{R}^{nm} \tag{10.12} \]

The main decision variables \( d_{(M,N)} \) is given as follows

\[ d_{(MX,N)} := (x_0^T, x_1^T, \ldots, x_N^T)^T \in \mathbb{R}^{N_1} \tag{10.13} \]

\[ d_{(MU,N)} := (u_0^T, u_1^T, \ldots, u_{N-1}^T)^T \in \mathbb{R}^{N_2} \tag{10.14} \]

\[ d_{(MK,N)} := k_k \in \mathbb{R}^{N_k} \tag{10.15} \]

\[ d_{(M,N)} := (d_{(MX,N)}^T, d_{(MU,N)}^T)^T \in \mathbb{R}^{N_1 + N_2 + N_k} \tag{10.16} \]

where

\[ N_1 = (N + 1)n + qn \sum_{k=1}^{N} k = (1 + \frac{qN}{2})(N + 1)n \tag{10.17} \]

\[ N_2 = Nm + qm \sum_{k=1}^{N-1} k = (1 + \frac{q(N-1)}{2})Nm \tag{10.18} \]

\[ N_3 = N_1 + N_2 + N_k \tag{10.19} \]

Constrain Slack Variables

The slack decision variable is given by

\[ d_{(S,N)} := (f^T, g^T, h^T)^T \in \mathbb{R}^{N_3} \tag{10.20} \]

with

\[ N_4 = \frac{(s + r)N(N - 1)}{2} + tN \]

The previous variables can be summarized as follows
\[ d_{(M,N)} := (d^T_{(M,N)}, d^T_{(S,N)})^T \in \mathbb{R}^{N_3+N_4=N_5} \]  

\textbf{Cost Slack Variables}

In order to formulate the desired LP we need to introduce cost slack variables. Let,

\[ a_0 := (a_{(0,0)}, a_{(0,1)}, \ldots, a_{(0,N)})^T \in \mathbb{R}^{N+1} \]  

\[ b_0 := (b_{(0,0)}, b_{(0,1)}, \ldots, b_{(0,N-1)})^T \in \mathbb{R}^N \]

Let for all \( k \in \mathbb{N}_{[1:N]} \) and all \( i \in \mathbb{N}_{[1:q]} \),

\[ a_{(i,k)} := (a_{(i,k,k)}, a_{(i,k,k+1)}, \ldots, a_{(i,k,N)})^T \in \mathbb{R}^{N+1-k} \]  

similarly for all \( k \in \mathbb{N}_{[1:N-1]} \) and all \( i \in \mathbb{N}_{[1:q]} \),

\[ b_{(i,k)} := (b_{(i,k,k)}, b_{(i,k,k+1)}, \ldots, b_{(i,k,N-1)})^T \in \mathbb{R}^{N-k} \]  

Furthermore, we introduce, \( k \in \mathbb{N}_{[1:N]} \)

\[ a_k := (a_{(1,k)}^T, a_{(2,k)}^T, \ldots, a_{(q,k)}^T)^T \in \mathbb{R}^{q(N+1-k)} \]  

similarly for all \( k \in \mathbb{N}_{[1:N-1]} \)

\[ b_k := (b_{(1,k)}^T, b_{(2,k)}^T, \ldots, b_{(q,k)}^T)^T \in \mathbb{R}^{q(N-k)} \]  

The cost slack variables \( d_{(C,N)} \) is given as follows

\[ d_{(CX,N)} := (a_0^T, a_1^T, \ldots, a_N^T)^T \in \mathbb{R}^{N_6} \]  

\[ d_{(CU,N)} := (b_0^T, b_1^T, \ldots, b_{N-1}^T)^T \in \mathbb{R}^{N_7} \]  

\[ d_{(C,N)} := (d_{(CX,N)}^T, d_{(CU,N)}^T)^T \in \mathbb{R}^{N_6+N_7} \]  

where
\[ N_0 = (N + 1) + q \sum_{k=1}^{N} k = (1 + \frac{qN}{2})(N + 1) \tag{10.31} \]

\[ N_1 = N + q \sum_{k=1}^{N-1} k = (1 + \frac{q(N - 1)}{2})N \tag{10.32} \]

\[ N_8 = N_6 + N_7 \tag{10.33} \]

Hence the overall decision variable \( d_N \) takes the following form

\[ d_{(M,N)} := (d_{(MS,N)}^T, d_{(C,N)}^T)^T \in \mathbb{R}^{N_9} \tag{10.34} \]

with \( N_9 = N_5 + N_8 \), where \( d_{(MS,N)} \) and \( N_5 \) are given as in equation 19.

**Equality Constraints**

The first set of equality constraints comes for the fact that \( X_{(k,k)} = W \):

\[ \forall k \in \mathbb{N}_{[1:q]} , x_{(i,k,k)} = w_i \tag{10.35} \]

The second set of equality constraints comes from the parametrize state tube dynamics

\[ \forall k \in \mathbb{N}_{[N-1]}, x_{(0,k+1)} = Ax_{(0,k)} + Bu_{(0,k)} \text{ and for } j \in \mathbb{N}_{[1:N-1]} \tag{10.36} \]

\[ \forall k \in \mathbb{N}_{[N-1]}, \forall i \in \mathbb{N}_{[1:q]}, x_{(i,j,k+1)} = Ax_{(i,j,k)} + Bu_{(i,j,k)} \tag{10.37} \]

\[ x_{(0,0,0)} = x_0 \tag{10.38} \]

**Bi-linear Constraints**

The equations that introduce bilinearity to the problem are the following

\[ \forall k \in \mathbb{N}_{[N-1]}, u_{(0,k)} = Kx_{(0,k)} \tag{10.39} \]
∀k ∈ N\([N-1]\), ∀i ∈ N\([1;q]\), u\(_{(i,j,k)}\) = Kx\(_{(i,j,k)}\) \quad (10.40)

where the feedback gain has the following structure

\[
K \in \mathbb{R}^{nm}, \quad K = \begin{bmatrix}
  k_{11} & k_{12} & k_{13} & \ldots & k_{1m} \\
  k_{21} & \vdots & \vdots & \ddots & \vdots \\
  k_{32} & \vdots & \vdots & \ddots & \vdots \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  k_{n1} & k_{n2} & k_{n3} & \ldots & k_{nm}
\end{bmatrix} \quad (10.41)
\]

for all \(i_2 \in R\), where \(i_2\) is the number of input variables,

∀k ∈ N\([1,N]\), ∀i_2 ∈ N\([1;m]\) Su\(_{(i_2,0,k)}\) = K\(_{(i_2,1:m)}\)x\(_{(0,k)}\) \quad (10.42)

∀i ∈ N\([1;q]\), ∀j ∈ N\([1:N-1]\), ∀k ∈ N\([1:N-1]\), ∀i_2 ∈ N\([1:n]\) Su\(_{(i,j,k)}\) = K\(_{(i_2,1:m)}\)x\(_{(i,j,k)}\) \quad (10.43)

this equation will lead to the following

\[
Sd_N = \bar{d}_N^T Hd_N \quad (10.44)
\]

The bilinear equations transform the problem into non-convex, non-linear optimization problem.

**Inequality Constraints**

In order to handle feasibility inequality constraints we have to introduce slack variables.

**State Constraints**

Let us consider the state constraints and introduce adequate set of slack variables as well as specify necessary constraints. Let for any \(i_1 \in N\([1,r]\):

f\(_{(i_1,1)}\) := f\(_{(i_1,1,1)}\) ∈ \mathbb{R} \quad (10.45)

f\(_{(i_1,2)}\) := (f\(_{(i_1,1,2)}\), f\(_{(i_1,2,2)}\)) ∈ \mathbb{R} \quad (10.46)
\[ f_{(i_1,3)} := (f_{(i_1,1,3)}, f_{(i_1,2,3)}, f_{(i_1,3,3)}) \in \mathbb{R} \]  

(10.47)

\[ f_{(i_1,N-1)} := (f_{(i_1,1,N-1)}, f_{(i_1,2,N-1)}, \ldots, f_{(i_1,3,N-1)}) \in \mathbb{R} \]  

(10.48)

and introduce for any \( i_1 \in \mathbb{N}_{[1:r]} \)

\[ f_{i_1} := (f_{(i_1,1)}, f_{(i_1,2)}, \ldots, f_{(i_1,N-1)}) \in \mathbb{R} \]  

(10.49)

and finally,

\[ f := (f_1^T, f_2^T, \ldots, f_r^T) \in \mathbb{R}^{r(N-1)/2} \]  

(10.50)

To satisfy state constraints we invoke the following set of conditions, for all \( i_1 \in \mathbb{N}_{[1:r]} \):

\[ F_{i_1}^T x \leq 1 \]

and for all \( k \in \mathbb{N}_{[1:N-1]} \)

\[ F_{i_1}^T x(0,0,k) + \sum_{j=1}^{k} f_{(i_1,j,k)} \leq 1 \]  

(10.52)

where for all \( k \in \mathbb{N}_{[1:N-1]} \) and all \( j \in \mathbb{N}_{[1:k]} \), \( \forall i \in \mathbb{N}_{[1:s]} \)

\[ F_{i_1}^T x(i,j,k) \leq f_{(i_1,j,k)} \]  

(10.53)

**Control Constraints**

Let for any \( i_1 \in \mathbb{N}_{[1:s]} \):

\[ g_{(i_1,1)} := g_{(i_1,1,1)} \in \mathbb{R} \]  

(10.54)
$$g(i_1,2) := (g(i_1,1,2), g(i_1,2,2)) \in \mathbb{R}^2 \quad (10.55)$$

$$g(i_1,3) := (g(i_1,1,3), g(i_1,2,3), g(i_1,3,3)) \in \mathbb{R}^3 \quad (10.56)$$

$$\vdots$$

$$g(i_1,N-1) := (g(i_1,1,N-1), g(i_1,2,N-1) \ldots \ldots, g(i_1,3,N-1)) \in \mathbb{R}^{N-1} \quad (10.57)$$

and introduce for any $$i_1 \in \mathbb{N}[1:s]$$

$$g_{i_1} := (g(i_1,1), g(i_1,2), \ldots \ldots, g(i_1,N-1)) \in \mathbb{R} \quad (10.58)$$

and finally,

$$g := (g_1^T, g_2^T, \ldots \ldots, g_s^T) \in \mathbb{R}^{sN(N-1)/2} \quad (10.59)$$

To satisfy control constraints we invoke the following set of conditions, for all $$i_1 \in \mathbb{N}[1:s]$$

$$G_{i_1}^T u(0,0) \leq 1$$

and for all

$$k \in \mathbb{N}[1:N-1]$$

$$G_{i_1}^T x_{(0,0,k)} + \sum_{j=1}^{k} g(i_1,j,k) \leq 1 \quad (10.61)$$

11/01/11

where for all $$k \in \mathbb{N}[1:N-1]$$ and all $$j \in \mathbb{N}[1:k]$$, $$\forall i \in \mathbb{N}[1:r]$$

$$G_{i_1}^T u(i,j,k) \leq g(i_1,j,k) \quad (10.62)$$

**Terminal Constraints**

Similar procedure applies to the terminal constraints. Let for any $$i_1 \in \mathbb{N}[1:r]$$:
\( h_{i_1} := (h_{i_1,1}, h_{i_1,2}, \ldots, h_{i_1,N}) \in \mathbb{R} \) \hspace{1cm} (10.63)

and finally,

\[ h := (h_1^T, h_2^T, \ldots, h_t^T) \in \mathbb{R}^tN \] \hspace{1cm} (10.64)

To satisfy terminal set constraints we invoke the following set of conditions, for all \( i_1 \in \mathbb{N}[1:t] \):

\[ H_{i_1}^T x_{(0,0,k)} + \sum_{j=1}^{k} h_{i_1,j} \leq 1 \] \hspace{1cm} (10.65)

where for all \( j \in \mathbb{N}[1:N], \forall i \in \mathbb{N}[1:q] \)

\[ H_{i_1}^T x_{(i,j,N)} \leq h_{i_1,j} \] \hspace{1cm} (10.66)

**Cost Constraints**

With the overall decision variables \( d_{(O,N)} \) we associate the \( O^{th} \) and \( (i,j)^{th} \) partial cost functions \( V_{(0,N)} \) and \( V_{(i,j,N)} : \mathbb{R}^{N_0} \rightarrow \mathbb{R}_+ \) given by

\[ V_{(0,N)}(d_{(O,N)}) := \sum_{k=0}^{N-1} l(x_{(0,k)}u_{(0,k)}) + V_f(x_{(0,N)}) \] \hspace{1cm} (10.67)

and

\[ V_{(i,j,N)}(d_{(O,N)}) := \sum_{k=0}^{N-1} l(x_{(i,j,k)}u_{(i,j,k)}) + V_f(x_{(i,j,N)}) \] \hspace{1cm} (10.68)

**Linear cost function**

To minimize the specified cost function via a linear programming we first introduce an appropriate set of cost reformulation related to constraints.

First we invoke the following constraints

\[ \forall l \in \mathbb{N}[N-1], \text{ } Qx_{(0,l)} \leq a_{(0,l)} \] \hspace{1cm} (10.69)

and\( P_x(0,N) \leq a_{(0,N)} \)

\[ \forall l \in \mathbb{N}[N], \text{ } -a_{(0,l)} \leq 0 \] \hspace{1cm} (10.70)
∀l ∈ N_{[N-1]}, \ R_u(0,l) ≤ b_{(0,l)} \quad (10.71)

∀l ∈ N_{[N]}, \ -b_{(0,l)} ≤ 0 \quad (10.72)

Second for the cost we invoke the following set of constraints for all \( k \in N_{[1:N-1]} \) and all \( i \in N_{[1,q]} \)

∀l ∈ N_{[N-1]}, \ Q_x(i,k,l) ≤ a_{(i,k,l)} \quad (10.73)

and \( P_x(i,k,N) \leq a_{(i,k,N)} \)

∀l ∈ N_{[N]}, \ -a_{(i,k,l)} ≤ 0 \quad (10.74)

∀l ∈ N_{[N-1]}, \ R_u(i,k,l) ≤ b_{(i,k,l)} \quad (10.75)

∀l ∈ N_{[N]}, \ -b_{(i,k,l)} ≤ 0 \quad (10.76)

Final, for \( k = N \) and for all \( i \in N_{[1,q]} \)

\( P_x(i,N,N) \leq a_{(i,N,N)} \quad (10.77) \)

and

\( -a_{(i,N,N)} ≤ 0 \quad (10.78) \)

**Min-max cost function**

The objective function can be formulated as follows

\[
\min_{\varphi_j} \sum_{k=0}^{j} \varphi_j
\]

\[ (10.79) \]

Additional constraints and slack variables should be included in order to introduce the min-max cost function.
\[
\left( \sum_{k=0}^{N-1} (x_{(0,k)}^T Q x_{(0,k)} + u_{(0,k)}^T R u_{(0,k)}) + x_{(0,N)}^T P x_{(0,N)} \right)^{1/2} \leq \varphi_0 \quad (10.80)
\]

\[
\left( \sum_{k=0}^{N-1} (x_{(i,j,k)}^T Q x_{(i,j,k)} + u_{(i,j,k)}^T R u_{(i,j,k)}) + x_{(i,j,N)}^T P x_{(i,j,N)} \right)^{1/2} \leq \varphi_j \quad (10.81)
\]

where \( \varphi_0 \) is a nonnegative scalar such that \( V_{(0,N)} d_{(O,N)} \leq \varphi_0^2 \) and \( \varphi_j \) is a nonnegative scalar such that \( V_{(j,N)} d_{(O,N)} \leq \varphi_0^2 \)

**Linear Feedback gain**

Similar procedure we can follow in order to compute a feasible linear feedback gain when \( x_0 \in \mathcal{X} \). For, this case the \( x_0 \) is not a point but the vertices of the closed polytopic set of \( \mathcal{X} \). The main changes will be the followings

\[
x_{(i,0)} := (x_{(i,0,0,0)}, x_{(i,0,0,1)}, \ldots, x_{(i,0,0,N)})^T \in \mathbb{R}^{(N+1)nq} \quad (10.82)
\]

\[
u_{(i,0)} := (u_{(i,0,0,0)}, u_{(i,0,0,1)}, \ldots, u_{(i,0,0,N)})^T \in \mathbb{R}^{Nmq} \quad (10.83)
\]

If there is a feasible solution, this will lead to a feasible linear feedback gain which could guarantee stability of the linear system with additive disturbances, but could not guarantee optimality of the input trajectory. The table bellow describes the structure of the problem.
In this case the designed problem is \( q \) times bigger than the problem described previously, for the minimal robust invariant set. However the computed gain will be feasible for every \( x_0 \in X \).
10.3.6. Model reduction techniques

Eventhough optimal control design (online or offline) is recognized as the best possible way to control a system, still there are application where are not used due to the computation complexity and lack of accurate models usable for control. Model reduction techniques can be an alternative to overcome the lack of accurate and simplified models, since they can generate accurate and simplified mathematical models of large scale and complex mathematical models. Although the generated model is compacted, it can still captures the essential dynamic behaviour of the original model/process. In the context of multi-parametric optimization and control, the use of these techniques is considered critical and very promising.

10.3.7. Applications

The significant advances in multi–parametric programming and multi–parametric MPC were followed by a number of important applications. In section 5 a dynamic 2 dimension mathematical model has been presented for the metal hydride bed reactor. This model can be used to investigate alternative reactor designs in order to achieve better heating/cooling of the device during the adsorption/desorption. This can be done by altering the boundary conditions of the problem which specify the geometry of the metal hydride. Also it would be ideal to incorporate the metal hydride in the PEM fuel cell system model in order to test the performance of this integrated system.

In sections 6, 7 and 8 has been presented several mathematical models for a PEM fuel cell stack and units. The developed models have shown the expected behaviour however, due to the lack of detailed experimental data, these models wasn’t able to validated extensively in the real operation. The experimental unit, which has been design during my studies will be able to be used for the full validation of the models, as well as for the testing of the designed controllers. The unit has been designed for 1kW NEDSTACK PEM fuel cell, however is able to be used for lower and higher voltage fuel cell devices and for that reason can be used as test lab for PEM cells within the range of 0.1-3kW.

Finally, in Appendix III a mathematical model for the tunnel kiln has been designed which is able to predict accurately the operation of the firing process of the furnace. This model can be the base for the design of mathe-
mational model for the preheating and cooling zone in order to describe the full operation. Moreover this model can be used for the design a 2 dimension mathematical model in order to estimate certain phenomena during the operation,

- The reaction of the bricks/tiles due to the humidity;
- The effect of the CO$_2$ in the quality of the products;
- The optimal packing of the tiles/bricks in order to achieve minimum consumption of the fuel

10.4. Publications

Chapter in books

- Pistikopoulos,E.N., Chinchuluun,A., Dominguez,L., Panos,C., Kouramas,K. Theoretical and algorithmic advances in multi-parametric programming and control. CMS special volume, (accepted)

Journal articles


Conference proceedings


• Arce, A., Panos, C., Bordons, C., Pistikopoulos, E.N. (2011) Design and Experimental Validation of an Explicit MPC Controller for Regulating Temperature in PEM Fuel Cell Systems, Accepted for presentation in the 18th IFAC World Congress, Milan, Italy, 2011

• Kouramas, K., Panos, C., Pistikopoulos, E.N., (2011) Algorithm for robust explicit/multi-parametric MPC in embedded control systems. Accepted for presentation in the 18th IFAC World Congress, Milan, Italy, 2011


Chemical Engineering, 2011


10.5. Acknowledgements

The financial support of EPSRC (Projects EP/I014640/1, EP/047017/1) and KAUST is gratefully acknowledged.
Bibliography


249


A. Appendix I-Matlab Code for Explicit Robust MPC by Dynamic Programming

In this chapter it is presented the general Matlab code for the algorithm presented in chapters 2 and 3. This code consist of two function, the FinalSolution and ParametricSolution.

A.1. Code for FinalSolution.m

The Matlab file FinalSolution.m is the the master file which defines the structure of the problem. It collects the parametric solution of all the steps and performs gauss elimination to compute the final solution.

```matlab
model.Ap = [0.7326 -0.0861; 0.1722 0.9909];
model.Bp = [0.0609 ; 0.0064 ];
model.Cp = [1 0; 0 1];
mpc.Q = [0 1.4142];
mpc.R = 0.01*eye;
mpc.uL = [-2. ];
mpc.uU = [2.0 ];
mpc.xL = [-100 ;-100];
mpc.xU = [100;100];
mpc.TerminalXlow = mpc.xL;
mpc.Xlow0 = [-1,-1]
mpc.Xmax0 = [1;1];
mpc.TerminalXmax = mpc.xU;
mpc.Ny = 2;
option.plot = 0;
option.minmaxCRInf = 0;
option.minmaxMinimal = 0;
option.percentageTheta = 1.2;
option.solver = 2;
option.normalize = 1;
numberofx = size(model.Ap,2);
```

265
numberofu=size(model.Bp,2);
numberofy=size(model.Cp,1);
SolutionMethod.merging=2;
lola=1;
if lola==1;
    [SolutionDp1, mpv]=ParametricSolution(model,mpc,option);
end
for lola1=1:length(SolutionDp1)
    for lola2=1:length(SolutionDp1(lola1).CRInf)
        SolutionDp(lola1).CRInf(lola2).X=[SolutionDp1(lola1).CRInf(lola2).X];
        SolutionDp(lola1).CRInf(lola2).cr.b=[SolutionDp1(lola1).CRInf(lola2).cr.b];
    end
end

hhh=0;
if hhh==1;
    for j=1:length(mpv.Step)
        %disp('Ploting figures for each Step')
        fprintf(' Figure for Step %d
', j)
        subplot(length(mpv.Step),1,j)
        plot(mpv.Step(j).Pn)
    end
end

hh=0;
if hh==1
    for j=1:length(SolutionDp)
        dispQPSolution(SolutionDp(j).CRInf)
    end
end

lvpo=1;
if lvpo==1
    for i=1:length(SolutionDp)
        for j=1:length(SolutionDp(i).CRInf)
            SolutionDp(i).CRInf(j).X=SolutionDp(i).CRInf(j).X(1:numberofu,:);
        end
    end
end

if SolutionMethod.merging==1
    for i=1:length(SolutionDp)
        [SolutionDp(i).CRInf]=mergeExplicitMPC(SolutionDp(i).CRInf);
    end
elseif SolutionMethod.merging==0;
    [SolutionDp(i).CRInf]=mergeExplicitMPC(SolutionDp(1).CRInf);
elseif SolutionMethod.merging==2;
    for i=1:length(SolutionDp)
        [SolutionDp(i).CRInf]=mergeRegionsBegos(SolutionDp(i).CRInf,10000);
    end
elseif SolutionMethod.merging==3;
$\text{SolutionDp}(i).\text{CRInf} = \text{mergeRegionsBegos}($SolutionDp(1).\text{CRInf}, 10000$)
end
for i=1:length($\text{SolutionDp}(1).\text{CRInf}$)
Region.$\text{CRInf}(i).X = $SolutionDp(i).\text{CRInf}(i).X$;
Region.$\text{CRInf}(i).\text{cr.A} = $SolutionDp(i).\text{CRInf}(i).\text{cr.A}$;
Region.$\text{CRInf}(i).\text{cr.b} = $SolutionDp(i).\text{CRInf}(i).\text{cr.b}$;
end
for j=1:mpc.Ny
$\text{numberofRegions}(j) = \text{length}($SolutionDp(j).\text{CRInf}$);
end
for j=1:mpc.Ny-1
kk=0;
clear CRInf
for i1=1:length(Region.$\text{CRInf}$)
for i2=1:$\text{numberofRegions}(j+1)$
kk=kk+1;
% values of $x_0$
CRInf(kk).First_A = Region.$\text{CRInf}(i1).X(:,1:$$\text{numberofx}) * \text{model.Ap}$;
CRInf(kk).Second_A = $\text{SolutionDp}(j+1).\text{CRInf}(i2).X(:,1:$$\text{numberofx})$;
% values of $u_0$
CRInf(kk).First_B = Region.$\text{CRInf}(i1).X(:,1:$$\text{numberofx}) * \text{model.Bp} - \text{eye}(j*$$\text{numberofu},j*$$\text{numberofu})$;
CRInf(kk).Second_B = $-\text{eye}(\text{numberofu},\text{numberofu}) \text{SolutionDp}(j+1).\text{CRInf}(i2).X(:,\text{numberofx}+1:end-1)$;
% values for the constant parameter
CRInf(kk).First_b = Region.$\text{CRInf}(i1).X(:,end)$;
CRInf(kk).Second_b = $\text{SolutionDp}(j+1).\text{CRInf}(i2).X(:,end)$;
% Ac for the constrains
CRInf(kk).cr.First_A = Region.$\text{CRInf}(i1).\text{cr.A}$;
CRInf(kk).cr.First_b = Region.$\text{CRInf}(i1).\text{cr.b}$;
%bc for the constrains
CRInf(kk).cr.Second_A = $\text{SolutionDp}(j+1).\text{CRInf}(i2).\text{cr.A}$;
CRInf(kk).cr.Second_b = $\text{SolutionDp}(j+1).\text{CRInf}(i2).\text{cr.b}$;
end
for k=1:kk;
$\text{sol(k).Matrix} = [\text{CRInf(k).Second_B CRInf(k).Second_A CRInf(k).Second_b}]$;
$\text{CRInf(k).First_B CRInf(k).First_A CRInf(k).First_b]}$;
end
%solution
for k=1:kk
sol(k).R = rref(sol(k).Matrix);
sol(k).ub = $[-\text{sol(k).R(numberofu+1:end,(end-numberofx):end-1)} -\text{sol(k).R(numberofu+1:end,end)}]$;
sol(k).ua = $[-\text{sol(k).R(1:numberofu,(end-numberofx):end-1)} -\text{sol(k).R(1:numberofu,end)}]$;
sol(k).X = [sol(k).ua ; sol(k).ub]
end
for k=1:kk
Ac1 = CRInf(k).cr.First_A * [\text{model.Ap + model.Bp * sol(k).ua(:,1:$$\text{numberofx})}]$;
bcl = CRInf(k).cr.First_b - CRInf(k).cr.First_A * \text{model.Bp * sol(k).ua(:,end)}$;
Ac2 = CRInf(k).cr.Second * sol(k).ub(:,1:numberofx) + CRInf(k).cr.Second * A(:,numberofx+1:end);
b2 = CRInf(k).cr.Second * b - CRInf(k).cr.Second * A(:,numberofx+1:end) * sol(k).ub(:,end);
sol(k).cr.A = [Ac1; Ac2];
sol(k).cr.b = [b2; bc2];
clear Ac1 Ac2 bc1 bc2
end
o = 1;
clear Region.CRInf;
Region.CRInf = [];
g = 0;
if VERBOSE == 1
j;
end
for kk = 1:length(sol)
if g == 1
[xopt, fval, lambda, exitflag, how] = mpxpLP(zeros(1,(numberofx))', sol(kk).cr.A, sol(kk).cr.b,[],[],0,1);
if exitflag < 0
kk = kk + 1;
continue
end
[redCR, idrem] = reduceCR(sol(kk).cr);
Region.CRInf(o).cr.A = redCR.A;
Region.CRInf(o).cr.b = redCR.b;
Region.CRInf(o).X = sol(kk).X;
if VERBOSE == 1
o = o + 1
end
else
cr = deRedundandize(newA, newb);
if isempty(cr)
kk = kk + 1;
continue
end
kk;
if VERBOSE == 1
end
Region.CRInf(o).cr.A = cr.A;
Region.CRInf(o).cr.b = cr.b;
Region.CRInf(o).X = sol(kk).X;
o = o + 1;
end
end
end
plotCRInf(Region.CRInf)
A.2. Code for ParametricSolution.m

The Parametricsolution.m function generates the critical regions for each step, starting from the time $t = N$ and sends the solution to the master file. In addition it performs debug tests to examine if the solution in each step is compact and closed. Finally, it eliminates the redundant values from the inequality constraints and the explicit solution in order to avoid numerical errors.

```matlab
function [SolutionDp, mpv]=ParametricSolution(model,mpc,option)
% ——————————————————-
% The solutionDp has the order U, Yk, Wk
% You need to transform the values from Z to U
% The only parameter are the X.
% Copyright (c) 2007 parostech.com
% All rights reserved
% ——————————————————-

global ZERO;
global DEBUG;
global VERBOSE;
if DEBUG
	if size(model.Ap,1)˜=size(mpc.Q,1)
		error(‘pouli mou’)
	end
	if size(model.Bp,2)˜=size(mpc.R,1)
		error(‘pouli mou’)
	end
	if size(model.Ap,1)˜=size(mpc.xU,1)
		error(‘pouli mou’)
	end
	if size(model.Ap,1)˜=size(mpc.xL,1)
		error(‘pouli mou’)
	end
	if size(model.Bp,2)˜=size(mpc.uU,1)& size(model.Bp,2)˜=size(mpc.uU,2)
		error(‘pouli mou’)
	end
	if size(model.Bp,2)˜=size(mpc.uL,1)& size(model.Bp,2)˜=size(mpc.uL,2)
		error(‘pouli mou’)
	end
	mpv=[];
	Bp=model.Bp;
	Cp=model.Cp;
	R=mpc.R;
	Q=mpc.Q;
	P=mpc.P;
	Ny=mpc.Ny;
uU=mpc.uU;
uL=mpc.uL;
```

269
xL=mpc.xL;
xU=mpc.xU;
numberofx=size(Ap,2);
numberofu=size(Bp,2);
numberofy=size(Cp,1);
mpv.numberofx=size(Ap,2);
mpv.numberofu=size(Bp,2);

%% name of Parameters
mpv.NumberSolutions=mpc.Ny;
for i=1:Ny
  namesTheta=[];
  aux=1;
  for j=1:numberofx
    namesT1{aux}=sprintf('x%d(t+%d)', j,Ny-i);
    aux = aux+1;
  end
  if i>=2
    aux=1;
    for k=1:i-1
      for j=1:numberofu
        namesT2{aux}=sprintf('u%d(t+%d)', j,(Ny-i+k));
        aux = aux+1;
      end
    end
  end
  if i==1
    mpv.Step(i).namesTheta=[namesT1 ];
  else
    mpv.Step(i).namesTheta=[namesT1 namesT2 ];
  end
  end
  namesVar=[];
  for i=1:Ny
    aux=1;
    for j=1:numberofu
      namesU{aux}=sprintf('u%d(t+%d)', j,(Ny-i));
      aux=aux+1;
    end
    Step(i).namesU=namesU;
  end
  for i=1:Ny
    mpv.Step(i).namesVar=[Step(i).namesU ];
  end
  for j=1:Ny
    mpv.Step(j).ParameterMatrix=zeros(length(mpv.Step(j).namesTheta),1);
    mpv.Step(j).VariableMatrix=zeros(length(mpv.Step(j).namesVar),1);
  end
  clear Wcon Gcon Econ H Q F A Scon cr qp tStart F1 a1 Wcon3 Wcon4 ;
  fprintf('Dynamic Programming Step: %d 
', j)
%% Generate General Matrices
Q = mpc.Q ;
R=mpc.R;
Q=diagonalP(Q,j,P);
G=GmatrixU(Ap,Bp,j);
Fa=Ap;
for i=2:j;
Fa=[Fa; Ap^i];
end
G1=G(:,1:numberofu);
F1=[Fa G(:,numberofu+1:end)];
a1=[F1];
f1=2*a1'*Q*G1;
H=2*[G1'*Q*G1+R];
mpv.Step(j).f1=f1;
mpv.Step(j).H=H;
%% u<umax constrains
Step(j).Gcon(1)=[eye(numberofu) zeros(numberofu,length(mpv.Step(j).namesVar)-numberofu)];
Step(j).Wcon(1)=[uU ];
Step(j).Econ(1)=zeros(size(Step(j).Wcon{1},1),length(mpv.Step(j).ParameterMatrix));
Step(j).Scon(1)=Step(j).Econ{1}+Step(j).Gcon{1}*pinv(H)*f1';
if size(Step(j).Gcon{1},2)~=length(mpv.Step(j).namesVar);...
size(Step(j).Gcon{1},1)~=size(Step(j).Wcon{1},1);size(Step(j).Econ{1},2)~=length(mpv.Step(j).namesTheta)
error('malakia psile');
end
%% -u<-umin
Step(j).Gcon{2}=[eye(numberofu) zeros(numberofu,length(mpv.Step(j).namesVar)-numberofu)];
Step(j).Wcon{2}=[uL];
Step(j).Econ{2}=zeros(size(Step(j).Wcon{1},1),length(mpv.Step(j).ParameterMatrix));
Step(j).Scon{2}=Step(j).Econ{2}+Step(j).Gcon{2}*pinv(H)*f1';
if size(Step(j).Gcon{1},2)~=length(mpv.Step(j).namesVar);...
size(Step(j).Gcon{2},2)~=length(mpv.Step(j).namesVar);...
size(Step(j).Gcon{2},1)~=size(Step(j).Wcon{1},1);size(Step(j).Econ{1},2)~=length(mpv.Step(j).namesTheta)
error('malakia psile');
end
%% x(t+N)<xmax
if j=1
Step(j).Wcon{3}=[];
for i=1:j
Step(j).Wcon{3}=[Step(j).Wcon{3};mpc.TerminalXmax];
end
Step(j).Gcon{3}=[G1 zeros(size(G1,1),length(mpv.Step(j).namesVar)-numberofu)] ;
Step(j).Econ{3}=[-F1 zeros(size(F1,1),length(mpv.Step(j).namesTheta)-size(F1,2)) ];
Step(j).Scon{3}=Step(j).Econ{3}+Step(j).Gcon{3}*pinv(H)*f1';
else
Step(j).Wcon{3}=[];
for i=1:j
Step(j).Wcon{3}=[Step(j).Wcon{3};xU];
end

Step(j).Gcon{3} = [G1 zeros(size(G1,1),length(mpv.Step(j).namesVar)-numberofu)];
Step(j).Econ{3} = [-F1 zeros(size(F1,1),length(mpv.Step(j).namesTheta)-size(F1,2))];
Step(j).Scon{3} = Step(j).Econ{3} + Step(j).Gcon{3} * pinv(H)*f1';
end
if isempty(xU);
Step(j).Gcon = [];
Step(j).Wcon = [];
Step(j).Econ = [];
end

%% -x(t+N)<-xmin constrains
if j==1
Step(j).Wcon{4} = [];
for i=1:j
Step(j).Wcon{4} = [Step(j).Wcon{4}; -mpc.TerminalXlow];
end
Step(j).Gcon{4} = [-G1 zeros(size(G1,1),length(mpv.Step(j).namesVar)-numberofu)];
Step(j).Econ{4} = [-F1 zeros(size(F1,1),length(mpv.Step(j).namesTheta)-size(F1,2))];
Step(j).Scon{4} = Step(j).Econ{4} + Step(j).Gcon{4} * pinv(H)*f1';
else
Step(j).Wcon{4} = [];
for i=1:j
Step(j).Wcon{4} = [Step(j).Wcon{4}; xL];
end
Step(j).Gcon{4} = [-G1 zeros(size(G1,1),length(mpv.Step(j).namesVar)-numberofu)];
Step(j).Econ{4} = [-F1 zeros(size(F1,1),length(mpv.Step(j).namesTheta)-size(F1,2))];
Step(j).Scon{4} = Step(j).Econ{4} + Step(j).Gcon{4} * pinv(H)*f1';
end
if isempty(xL);
Step(j).Gcon{4} = [];
Step(j).Wcon{4} = [];
Step(j).Econ{4} = [];
end

%% previous step constrains (Step 1 doesnt have!!)
if j>2
if option.minmaxCRInf==1
%% x<xmax
Step(j).Gcon{6} = [Bp ];
Step(j).Wcon{6} = mpc.Step(j-1).Solution.maxX(1:numberofx);
Step(j).Econ{6} = [-Ap zeros(numberofx,length(mpv.Step(j).namesTheta)-numberofx)];
Step(j).Scon{6} = Step(j).Econ{6} + Step(j).Gcon{6} * pinv(H)*f1';

%% x>xmin
Step(j).Gcon{7} = [-Bp ];
Step(j).Wcon{7} = -mpc.Step(j-1).Solution.minX(1:numberofx);;
Step(j).Econ{7} = [Ap zeros(numberofx,length(mpv.Step(j).namesTheta)-numberofx)];
Step(j).Scon{7} = Step(j).Econ{7} + Step(j).Gcon{7} * pinv(H)*f1';
end
Step(j).Gcon{5} = mpc.Step(j-1).Solution.matrixA(:,1:numberofx)*[Bp ];
Step(j).Wcon{5} = mpc.Step(j-1).Solution.matrixb;
mpv.Step(j-1).Solution.matrixA(:,1:numberofx)*zeros(numberofx,numberofu) -
mpv.Step(j-1).Solution.matrixA(:,numberofx+1:end)];
Step(j).Scon{5} = Step(j).Econ{5} + Step(j).Gcon{5} * pinv(H)*f1';
end

%% x(0)<xmax
Step(j).Gcon{8} = zeros(numberofx,length(mpv.Step(j).namesVar));
Step(j).Wcon{8} = mpc.Xmax0;
Step(j).Econ{8} = [-eye(numberofx) zeros(numberofx,length(mpv.Step(j).namesTheta)-numberofx)];
Step(j).Scon{8} = Step(j).Econ{8} + Step(j).Gcon{8} * pinv(H)*f1';

%% x(0)<xmin
Step(j).Gcon{9} = zeros(numberofx,length(mpv.Step(j).namesVar));
Step(j).Wcon{9} = -mpc.Xlow0;
Step(j).Econ{9} = [eye(numberofx) zeros(numberofx,length(mpv.Step(j).namesTheta)-numberofx)];
Step(j).Scon{9} = Step(j).Econ{9} + Step(j).Gcon{9} * pinv(H)*f1';

%% Overall Constrains
Gcon_total=[];
Wcon_total=[];
Econ_total=[];
Scon_total=[];
for i=1:length(Step(j).Gcon)
Gcon_total=[Gcon_total; Step(j).Gcon{i} ];
Wcon_total=[Wcon_total; Step(j).Wcon{i} ];
Econ_total=[Econ_total; Step(j).Econ{i} ];
Scon_total=[Scon_total; Step(j).Scon{i} ];
end

%% the last part of Step(j).Gcon(end), Wcon, Econ is the overall!
Step(j).Gcon{length(Step(j).Gcon)+1} = Gcon_total;
Step(j).Wcon{length(Step(j).Wcon)+1} = Wcon_total;
Step(j).Econ{length(Step(j).Econ)+1} = Econ_total;
Step(j).Scon{length(Step(j).Scon)+1} = Scon_total;

%% normalize the values to ZERO!
if option.normalize==1
for k1=1:size(Gcon_total,1)
for k2=1:size(Gcon_total,2)
if Gcon_total(k1,k2)<ZERO & Gcon_total(k1,k2)>0
Gcon_total(k1,k2)=0;
end
if Gcon_total(k1,k2)<0 & Gcon_total(k1,k2)>ZERO
Gcon_total(k1,k2)=0;
end
end
for k1=1:size(Wcon_total,1)
for k2=1:size(Wcon_total,2)
if Wcon_total(k1,k2)<ZERO & Wcon_total(k1,k2)>0
Wcon_total(k1,k2)=0;
end
end
end

273
end
if Wcon_total(k1,k2)<0 & Wcon_total(k1,k2)>-ZERO
  Wcon_total(k1,k2)=0;
end
end
end
for k1=1:size(Econ_total,1)
  for k2=1:size(Econ_total,2)
    if Econ_total(k1,k2)<ZERO & Econ_total(k1,k2)>0
      Econ_total(k1,k2)=0;
    end
    if Econ_total(k1,k2)<0 & Econ_total(k1,k2)>-ZERO
      Econ_total(k1,k2)=0;
    end
  end
end
for k1=1:size(Scon_total,1)
  for k2=1:size(Scon_total,2)
    if Scon_total(k1,k2)<ZERO & Scon_total(k1,k2)>0
      Scon_total(k1,k2)=0;
    end
    if Scon_total(k1,k2)<0 & Scon_total(k1,k2)>-ZERO
      Scon_total(k1,k2)=0;
    end
  end
end
mpv.Step(j).Gcon=Gcon_total;
mpv.Step(j).Wcon=Wcon_total;
mpv.Step(j).Econ=Econ_total;
mpv.Step(j).Scon=Scon_total;
for i=1:length(Step(j).Gcon)
  if size(Step(j).Gcon{i},1)~=size(Step(j).Wcon{i},1)...
    size(Step(j).Gcon{i},1)~=size(Step(j).Econ{i},1)
    error('malakies sta constrains')
  else
    if i<length(Step(j).Gcon) & DEBUG_ON==2
      fprintf('Correct constrain %d \n', i)
    elseif i==length(Step(j).Gcon) & DEBUG_ON==2
      fprintf('Correct total constrain \n')
    else
      continue
    end
  end
end
Step(j).Scon{end+1}=Step(j).Econ{end}+Step(j).Gcon{end}*inv(H)*f1';
if (Step(j).Scon{end}-Step(j).Scon{end-1})>ZERO
  error('Error in S constrains')
end
end
end
Q=H;
c=zeros(length(mpv.Step(j).namesVar),1);
A=Step(j).Gcon{end};
b=Step(j).Wcon{end};
F=Step(j).Scon{end};

%% Parameters Bounds

if isempty(mpc.xU) & isempty(mpc.uU)
tUp=200*ones(1,length(mpv.Step(j).namesTheta))';
else
tUp=[option.percentageTheta*xU'];
for i=2:j
tUp=[tUp option.percentageTheta*uU'];
end
end
tUpt=tUp';
if length(tUpt)=length(mpv.Step(j).namesTheta)
error('malakia sta tUp! checkare')
end
tLw=[];
if isempty(mpc.xU) & isempty(mpc.uU)
tLw=-200*ones(1,length(mpv.Step(j).namesTheta))';
else
tLw=[option.percentageTheta*xL'];
for i=2:j
tLw=[tLw option.percentageTheta*uL'];
end
end
tLwt=tLw';
if length(tLwt)=length(mpv.Step(j).namesTheta)
error('malakia sta tLw! checkare')
end
tU = zeros(size(tUp,1),1); % no theta in the objective neither
cc = 0;

%% Normalize Constrains

if 1
nx = size(A,2);
[A b kak] = normalizeConstraints([A F], b, [], []);
F = A(:,nx+1:end);
A = A(:,1:nx);
end

%% mpQP structure
qp = struct('cx', c, 'ct', ct, 'cc', cc,'Q', Q,'A', A, 'b', b, 'F', F, ...
'D', [], 'e', [], 'H', []);
cr = bound2Constr(tLw, tUp);
cr2=deRedundandize([A F], b);

275
if option.solver==1
  tStart=(1.1*tLow+tUp)/2;
  disp(‘———–solver 1—————’)
  [CRInf, infes] = mpQP4(qp, cr,tStart);
  if length(CRInf)==0;
    disp(‘have to change to solver 2’) option.solver=2;
  end
elseif option.solver==2
  disp(‘———–solver 2—————’)
  DEBUG_ON=0;
  [CRInf, Pn] = mpQP3(qp, cr);
  mpv.Step(j).Pn=Pn;
else
  error(‘option.solver can have values 1 or 2’) end
for ik=1:length(CRInf);
  SolutionDp(j).CRInf(ik) = CRInf(ik);
  SolutionDp(j).CRInf(ik).X = (CRInf(ik).X - [inv(H)*f1' zeros(length(mpv.Step(j).namesVar),1)]);
end
for ik=1:length(CRInf);
  mpv.Step(j).CRInf(ik).cr = CRInf(ik).cr;
end
%% Normalize values to ZERO
if option.normalize==1
  for k1=1:length(SolutionDp(j).CRInf)
    for k2=1:size(SolutionDp(j).CRInf(k1).X,1)
      for k3=1:size(SolutionDp(j).CRInf(k1).X,2)
        if SolutionDp(j).CRInf(k1).X(k2,k3) < ZERO & SolutionDp(j).CRInf(k1).X(k2,k3) > 0
          SolutionDp(j).CRInf(k1).X(k2,k3)=0;
        end
        if SolutionDp(j).CRInf(k1).X(k2,k3) < 0 & SolutionDp(j).CRInf(k1).X(k2,k3) >-ZERO
          SolutionDp(j).CRInf(k1).X(k2,k3)=0;
        end
      end
    end
  end
  for k1=1:length(SolutionDp(j).CRInf)
    for k2=1:size(SolutionDp(j).CRInf(k1).cr.A,1)
      for k3=1:size(SolutionDp(j).CRInf(k1).cr.A,2)
        if SolutionDp(j).CRInf(k1).cr.A(k2,k3) < ZERO & SolutionDp(j).CRInf(k1).cr.A(k2,k3) > 0
          SolutionDp(j).CRInf(k1).cr.A(k2,k3)=0;
        end
        if SolutionDp(j).CRInf(k1).cr.A(k2,k3) < 0 & SolutionDp(j).CRInf(k1).cr.A(k2,k3) >-ZERO
          SolutionDp(j).CRInf(k1).cr.A(k2,k3)=0;
        end
      end
    end
  end
end

276
for k1=1:length(SolutionDp(j).CRInf)
    for k2=1:size(SolutionDp(j).CRInf(k1).cr.b,1)
        for k3=1:size(SolutionDp(j).CRInf(k1).cr.b,2)
            if SolutionDp(j).CRInf(k1).cr.b(k2,k3) < ZERO & SolutionDp(j).CRInf(k1).cr.b(k2,k3) > 0
                SolutionDp(j).CRInf(k1).cr.b(k2,k3) = 0;
            end
            if SolutionDp(j).CRInf(k1).cr.b(k2,k3) < 0 & SolutionDp(j).CRInf(k1).cr.b(k2,k3) > -ZERO
                SolutionDp(j).CRInf(k1).cr.b(k2,k3) = 0;
            end
        end
    end
end

minmaxCRInf(SolutionDp(j).CRInf, option.minmaxCRInf, option.minmaxMinimal, mpc, numberofx);
mpv.Step(j).Solution.minX = SolutionDp(j).minX;
mpv.Step(j).Solution.maxX = SolutionDp(j).maxX;
mpv.Step(j).Solution.matrixb = SolutionDp(j).matrixb;
end

%% Eliminate Values Less than 10E-10
hk = 1;
if hk == 1;
    for j = 1:Ny
        for i = 1:length(SolutionDp(j).CRInf)
            for k = 1:length(SolutionDp(j).CRInf(i).X)
                if SolutionDp(j).CRInf(i).X(k) <= 10^(-10) & SolutionDp(j).CRInf(i).X(k) >= 0;
                    SolutionDp(j).CRInf(i).X(k) = 0;
                end
                if SolutionDp(j).CRInf(i).X(k) >= 10^(-10) & SolutionDp(j).CRInf(i).X(k) <= 0;
                    SolutionDp(j).CRInf(i).X(k) = 0;
                end
            end
        end
    end
end
fff = 0;
if fff == 1
    for j = 1:length(SolutionDp)
        for i = 1:length(SolutionDp(j).CRInf(i))
            SolutionDp(j).CSR(i) = SolutionDp(j).CRInf(i);
        end
    end
end
end
if option.plot == 1;
    for j = 1:Ny
        figure(j)
        if j == 1 & ;
            plotCRInfo2D(SolutionDp(j).CRInf)
else
fprintf('Plot %d',j)
tfixed=[NaN NaN 0*ones(1,length(mpv.Step(j).namesTheta)-2)]'
plot(CRinfoSlice(SolutionDp(j).CRInf, tfixed)
end
end
disp('End of Parametric Dynamic Programming')
disp('———————————') end
function [OO]=diagonalP(A, Ny,P)
%Generates diagonal matrix from given matrix A and given size of the
OO=[];
inca=A;
x=size(A,1);
y=size(A,2);
row1=zeros(x,y*Ny);
row2=zeros(x,y*Ny);
for i=1:Ny;
if i==Ny
inc=P;
end
R1=row1(:,1:(Ny-i)*y);
R2=row2(:,1:(i-1)*y);
OO=[OO; R2 inc R1];
row1=R1;
end
OO;
end
function [G]=GmatrixU(A,B,Ny);
numberofx=size(A,2);
numberofu=size(B,2);
G=[];
inca=B;
row=zeros(numberofx,(Ny)*numberofu);
for i=1:(Ny); R1=row(:,1:((Ny)-1)*numberofu);
R1=[inc, R1];
inc=A*inc;
G=[G; R1];
row=R1;
end
end
function [minX, maxX, matrixA, matrixb ]=minmaxCRInf(CRInf,option,minmaxMinimal,mpc,numberofx);
global ZERO
if minmaxMinimal==0;
method=1;
if method==0
[CRInf,PP]=UnionBegosNew3(CRInf)
end
for i=1:length(CRInf)
    area(i).K=CRInf(i).cr.b;
    area(i).H=CRInf(i).cr.A;
    poly(i)=polytope(area(i));
end
Options.lpsolver=0;
Options.verbose=0;
pp=union(poly,Options);
if length(pp)˜=1;
    [CRInf,PP]=AdvanceUnion(CRInf,'all')
else
    minmaxMinimal=1;
    for i=1:length(CRInf)
        area(i).K=CRInf_new(i).cr.b;
        area(i).H=CRInf_new(i).cr.A;
        poly(i)=polytope(area(i));
    end
end
matrixA matrixb=double(pp);
numberOfX=size(matrixA,2);
minX=[];
maxX=[];
if option==1
    for i=1:numberofX
        OptimalMatrixMin=[zeros(1,i-1) 1 zeros(1,numberOfX-i)] ;
        newMin=linprog(OptimalMatrixMin,matrixA,matrixb);
        minX=[minX ; newMin(i)];
    end
    for i=1:numberofX
        OptimalMatrixMax=[zeros(1,i-1) -1 zeros(1,numberOfX-i)] ;
        newMax=linprog(OptimalMatrixMax,matrixA,matrixb);
        maxX=[maxX ; newMax(i)];
    end
    for i=1:numberofX
        if minX(i)<ZERO & minX(i)>0
            minX(i)=0;
        end
    end
    for i=1:numberofX
        if maxX(i)<ZERO & maxX(i)>0
            maxX(i)=0;
        end
    end
    for i=1:numberofX
        if minX(i)>ZERO & minX(i)<0
            minX(i)=0;
        end
    end
end
279
end
end
for i=1:numberofX
if maxX(i)>-ZERO & maxX(i)<0
maxX(i)=0;
end
end
end
end
B. Appendix II-PEM Fuel Cell Lab

B.1. Unit Specifications

The main unit specifications for the fuel, cooling system, electrical and mechanical design, as well as the operating conditions can be summarized in the following tables.

Table B.1.: Electrical design

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power max</td>
<td>≥ 1000 W_e</td>
</tr>
<tr>
<td>Voltage</td>
<td>4.4 V</td>
</tr>
<tr>
<td>Voltage max</td>
<td>8 V</td>
</tr>
<tr>
<td>Current max</td>
<td>230 A</td>
</tr>
</tbody>
</table>

Table B.2.: Cooling System

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>1-2 kW</td>
</tr>
<tr>
<td>Medium</td>
<td>De-mineralized water</td>
</tr>
<tr>
<td>Purity</td>
<td>Conductivity &lt; 17 µSiemens</td>
</tr>
<tr>
<td>Pressure difference</td>
<td>&lt; 0.5 bar</td>
</tr>
<tr>
<td>Operating window</td>
<td>DT &lt; 10 K</td>
</tr>
</tbody>
</table>

Table B.3.: Mechanical design

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>15 Kg</td>
</tr>
<tr>
<td>Size</td>
<td>265x183x263 mm</td>
</tr>
</tbody>
</table>
### Table B.4: Fuel specifications

<table>
<thead>
<tr>
<th>Hydrogen</th>
<th>100% RH at 60°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purity</td>
<td>Grade 5</td>
</tr>
<tr>
<td>Pressure drop</td>
<td>&lt;100mbar at max power</td>
</tr>
<tr>
<td>Stoichiometry</td>
<td>&gt;1.25 for H2</td>
</tr>
<tr>
<td>Anode flow</td>
<td>&gt;5 Nl/min</td>
</tr>
</tbody>
</table>

### Table B.5: Air specifications

<table>
<thead>
<tr>
<th>Filtered</th>
<th>100% RH at 60°C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purity</td>
<td>100% RH at 60°C</td>
</tr>
<tr>
<td>Pressure drop</td>
<td>&lt;200mbar at max power</td>
</tr>
<tr>
<td>Stoichiometry</td>
<td>&gt;1.25 for air</td>
</tr>
<tr>
<td>Cathode flow</td>
<td>&gt;8 Nl/min</td>
</tr>
</tbody>
</table>

### B.2. List of materials

In the following table is presented the complete list of material for the design PEM fuel cell unit.

#### Table B.7: List of materials

<table>
<thead>
<tr>
<th>No</th>
<th>ITEM</th>
<th>DESCRIPTION</th>
<th>MANUFACTURER</th>
<th>MODEL No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CK-21A/ CK-81/ CK-82</td>
<td>Check valve 3/8”</td>
<td>Swagelok</td>
<td>SS-CHS6-1</td>
</tr>
<tr>
<td>2</td>
<td>CK-21C/ CK-22C/ CK-91</td>
<td>Check valve 3/4”</td>
<td>Swagelok</td>
<td>SS-CHS12-1</td>
</tr>
<tr>
<td>3</td>
<td>CK-21B</td>
<td>Check valve 1/4”</td>
<td>Swagelok</td>
<td>SS-CHS4-1</td>
</tr>
<tr>
<td>4</td>
<td>CT-51</td>
<td>Conductivity Electrode</td>
<td>JUMO</td>
<td>ecoTRANS Lf 01 Type 202731/01-015/024</td>
</tr>
<tr>
<td>5</td>
<td>F-21A</td>
<td>All welded in line filter 3/8”</td>
<td>Swagelok</td>
<td>SS-6F-7</td>
</tr>
<tr>
<td>6</td>
<td>F-21B</td>
<td>All welded in line filter 1/4”</td>
<td>Swagelok</td>
<td>SS-4FW-7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>7</td>
<td>F-21C</td>
<td>All welded in line filter 1/2”</td>
<td>Swagelok</td>
<td>SS-8F-7</td>
</tr>
<tr>
<td>8</td>
<td>F-51</td>
<td>Water Filter</td>
<td>Parker</td>
<td>HIFCE01AAN-Z-E</td>
</tr>
<tr>
<td>9</td>
<td>F-52</td>
<td>Water Filter</td>
<td>HERCO-DESTILO</td>
<td>D2</td>
</tr>
<tr>
<td>10</td>
<td>FN-51 V-51 HE-51</td>
<td>Radiator FAN</td>
<td>Renault</td>
<td>LCV Extra Mk1 86-91 1.1</td>
</tr>
<tr>
<td>11</td>
<td>HV-21C HV-93</td>
<td>Hand operated 2-way ball valve 3/4”</td>
<td>Swagelok</td>
<td>SS-45S12</td>
</tr>
<tr>
<td>12</td>
<td>HV-21C HV-93 HV-21A HV-83</td>
<td>Hand operated 2-way ball valve 3/8”</td>
<td>Swagelok</td>
<td>SS-44S6</td>
</tr>
<tr>
<td>13</td>
<td>HV-21B HV-31 HV-32 HV-81 HV-91</td>
<td>Hand operated 2-way ball valve 1/4”</td>
<td>Swagelok</td>
<td>SS-43GS4</td>
</tr>
<tr>
<td>14</td>
<td>JE-52 JE-82 JE-92</td>
<td>Heat tracing</td>
<td>OMEGA</td>
<td>STH052-040LSE</td>
</tr>
<tr>
<td>15</td>
<td>MFC-21A</td>
<td>Thermal mass flow meter H2 (28l/min)</td>
<td>Bronkhorst</td>
<td>F-201AV-50K-AFD-00-V</td>
</tr>
<tr>
<td>16</td>
<td>MFC-21B</td>
<td>Thermal mass flow meter N2 (10l/min)</td>
<td>Bronkhorst</td>
<td>F-201CV-10K-AFD-22-V</td>
</tr>
<tr>
<td>17</td>
<td>MFI-21C</td>
<td>Thermal mass flow meter AIR (70l/min)</td>
<td>Bronkhorst</td>
<td>F-111AC-50K-AFD-55-V</td>
</tr>
<tr>
<td>18</td>
<td>P-21</td>
<td>Air Compressor</td>
<td>DOMEL</td>
<td>497.3.265-854</td>
</tr>
<tr>
<td>19</td>
<td>P-51</td>
<td>Pump (Cooling System)</td>
<td>Cole Parmer</td>
<td>RZ-07002-65</td>
</tr>
<tr>
<td>20</td>
<td>P-81 P-91</td>
<td>Hydrator Pump</td>
<td>SEKO</td>
<td>PE-1.3-FP-T</td>
</tr>
<tr>
<td>21</td>
<td>P-82</td>
<td>Hydrogen Pump</td>
<td>Cole Parmer</td>
<td>RZ-79200-10</td>
</tr>
</tbody>
</table>

283
<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>PCV-83</td>
<td>Pressure Regulator 1/2”</td>
<td>TESCOM</td>
<td>6-1-B-2-12-H-G-A</td>
</tr>
<tr>
<td>23</td>
<td>PCV-93</td>
<td>Pressure Regulator 3/4”</td>
<td>TESCOM</td>
<td>6-1-B-2-08-H-G-A</td>
</tr>
<tr>
<td>24</td>
<td>PDT-101</td>
<td>Different Pressure Transmitter</td>
<td>FOXBORO</td>
<td>IDP10-T22B1N</td>
</tr>
<tr>
<td>25</td>
<td>PI-21A PI-21B PI-21C PI-51 PI-91 PI-81</td>
<td>Pressure gauge</td>
<td>WIKA</td>
<td>233.50.63</td>
</tr>
<tr>
<td>26</td>
<td>PSV-21A PSV-21C</td>
<td>Relief valve 1/2”</td>
<td>Swagelok</td>
<td>SS-RL4S8</td>
</tr>
<tr>
<td>27</td>
<td>PSV-21B</td>
<td>Relief valve 1/4”</td>
<td>Swagelok</td>
<td>SS-4R3A</td>
</tr>
<tr>
<td>28</td>
<td>PT-51 PT-82 PT-83 PT-92 PT-93</td>
<td>Pressure transducer 1/4”</td>
<td>JUMO</td>
<td>404366/000-458-402-511-20-61</td>
</tr>
<tr>
<td>29</td>
<td>R-101</td>
<td>PEM Fuel Cell stack</td>
<td>NEDSTACK</td>
<td>S008AAH0694</td>
</tr>
<tr>
<td>30</td>
<td>TE-51 TE-52 TE-82 TE-83 TE-92 TE-93</td>
<td>Thermocouple</td>
<td>OMEGA</td>
<td>KTIN-116U-12</td>
</tr>
<tr>
<td>31</td>
<td>V-52</td>
<td>Water Feed Vessel Plastic 10Lit</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>32</td>
<td>V-81(LS-81) (TE-81) (JE-81)</td>
<td>Hydrator vessel 28SLM</td>
<td>Fuel Cell Technologies</td>
<td>-</td>
</tr>
<tr>
<td>33</td>
<td>V-82 V-92</td>
<td>Dillution vessel</td>
<td>Swagelok</td>
<td>304L-HDF8-1000</td>
</tr>
<tr>
<td>34</td>
<td>LS-92</td>
<td>Level transmitter</td>
<td>FOXBORO</td>
<td>IDP10-T22B1N</td>
</tr>
<tr>
<td>35</td>
<td>V-91(LS91) (TE-91) (JE-91)</td>
<td>Hydrator vessel 70SLM</td>
<td>Fuel Cell Technologies</td>
<td>-</td>
</tr>
<tr>
<td>36</td>
<td>XT-101</td>
<td>Electronic Load</td>
<td>EA</td>
<td>EL9000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Interface Card RS232</td>
<td>EA</td>
<td>IF-R1</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>XV-21A XV-81 XV-83</td>
<td>2-way Solenoid valves 3/8” NC</td>
<td>SKINNER</td>
<td>71215SN33N00</td>
</tr>
<tr>
<td>39</td>
<td>XV-82 XV93</td>
<td>2-way Solenoid valve 3/8” NO</td>
<td>SKINNER</td>
<td>72228RN3TV00</td>
</tr>
<tr>
<td>40</td>
<td>XV-21B XV-22C</td>
<td>2-way Solenoid valves 1/4” NO</td>
<td>SKINNER</td>
<td>71225SN2GF00</td>
</tr>
<tr>
<td>41</td>
<td>XV-92</td>
<td>2-way Solenoid valve 3/4” NO</td>
<td>SKINNER</td>
<td>72228RN5VV00</td>
</tr>
<tr>
<td>42</td>
<td>XV-21C XV-91</td>
<td>2-way Solenoid valves 3/4” NC</td>
<td>SKINNER</td>
<td>72218RN5VV00</td>
</tr>
<tr>
<td>43</td>
<td>XV-22A</td>
<td>3-way Solenoid valve 3/8”</td>
<td>SKINNER</td>
<td>73312BN3RNJ0</td>
</tr>
<tr>
<td>44</td>
<td>Hydrogen detector</td>
<td>HDGI</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

**B.3. PEM Fuel Cell unit design**

**B.3.1. Frame and Panel**

In this section is presented the three dimension figures for the frame and the unit.

**B.3.2. Control Cabinet**

The control cabinet will control the electrical devices of the unit.
Figure B.1.: 3D Frame Layout

Figure B.2.: Frame Layout
Figure B.3.: Panel Layout

Figure B.4.: Unit Layout
Figure B.5.: Unit Layout

Figure B.6.: Control cabinet design
## Table B.6.: Stack operating conditions

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Temperature</strong></td>
<td>65°C</td>
</tr>
<tr>
<td><strong>Pressure</strong></td>
<td>Atmospheric</td>
</tr>
<tr>
<td><strong>Stack outlets pressure</strong></td>
<td>Ambient</td>
</tr>
</tbody>
</table>

## Figure B.7.: External view of the cabinet