OVERALL INTELLIGENT HYBRID CONTROL SYSTEM
FOR A FOSSIL-FUEL POWER UNIT

A Thesis in
Electrical Engineering

by

Raul Garduno-Ramirez

© 2000 Raul Garduno-Ramirez

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2000
We approve the thesis of Raul Garduno-Ramirez.

________________________                                                            _________________
Kwang Y. Lee
Professor of Electrical Engineering
Thesis Advisor
Chair of Committee

________________________
Mario Sznaier
Associate Professor of Electrical Engineering

________________________
Jeffrey S. Mayer
Associate Professor of Electrical Engineering

________________________
Constantino M. Lagoa
Assistant Professor of Electrical Engineering

________________________
Robert M. Edwards
Associate Professor of Nuclear Engineering

________________________
W. Kenneth Jenkins
Professor of Electrical Engineering
Head of the Department of Electrical Engineering
ABSTRACT

The current global problem scenario faced by power plants is characterized by a changing multiplicity of everyday-tighter operation requirements (e.g., life extension, pollution regulation, cyclic operation, heat rate improvement, etc.). In this situation, an overall approach for optimal operation and control of power units becomes of paramount relevance for the survival of utilities competing under liberalized generation markets. In response to these circumstances, this dissertation contributes a methodology to design a generalized overall unit control system for a fossil fuel power unit (FFPU), and develops a minimum prototype to demonstrate its feasibility.

Toward the above goal, the associated research project was undertaken as a technology innovation process with its two ends identified as follows. First, it is recognized that the coordinated control strategies constitute the uppermost control level in current FFPPUs, and so, are responsible for driving the boiler-turbine-generator set as a single entity. Second, a FFPU is envisioned as a complex process, subject to multiple changing operating conditions, that should perform as an intelligent system, for which an advanced integral control concept is needed. Therefore, as an outcome of the innovation process, a generalized unit control concept that extends the capabilities of current coordinated control schemes is proposed. This concept is presented as the Intelligent Coordinated Control System (ICCS) paradigm, which establishes an open reference framework for the development of overall unit control schemes.

The ICCS realizes a multi-agent system for which a multidisciplinary approach, that amalgamates control, process, and software engineering concepts, is established for its design. The ICCS’s system goals are identified using power plant process engineering concepts, and intelligent control systems engineering concepts are used to identify main tasks and to achieve system functional decomposition. A software engineering agency concept is used to identify and group agents according to their knowledge and purpose interactions. The resultant ICCS structure is an open set of functionally grouped agent clusters in a two-level hierarchical system. The upper level, mainly characterized for
knowledge-driven processes, performs the supervisory functions needed to provide self-governing operation characteristics, while the lower level, mainly characterized for data-driven processes, performs the fast reactive behavior functions necessary for hybrid real-time control and protection.

The Minimum Prototype of the Intelligent Coordinated Control System (ICCS-MP) comprehends a minimum set of functions needed by the power unit to participate in the total automation of power systems. Basically, the ICCS-MP provides the means to achieve optimized wide-range cyclic operation, by being able to follow any given unit load demand profile issued by upper level economic dispatch and unit commitment agents, and to optimally accommodate an arbitrary number of generally conflicting operating objectives.

Developed through several stages, the ICCS-MP finally implements a two-level hierarchical intelligent hybrid multi-agent coordinated control system. The supervisory functions include optimization and command generation, learning and control tuning, and performance and state monitoring. The direct level consist of a multivariable feedforward and feedback control scheme. The implementation core of the system is formed by three modules: reference governor, feedforward control processor, and feedback control processor. The performance and state monitoring, and learning and control tuning functions can be executed either under demand in an off-line basis or are implicitly included in the main modules.

The reference governor generates set-point trajectories for the lower level control loops by solving a multiobjective optimization problem, for which the objective functions and their priorities can be set arbitrarily, in number and form. This approach allows for process optimization, and provides a way to specify the operating policy to accommodate a great diversity of operating scenarios. The proposed feedforward-feedback control scheme is an extension of the general linear single-input-single-output feedback control scheme, with both reference feedforward and disturbance feedforward actions, to the nonlinear multivariable case. The feedforward control processor is implemented using a set of multi-input-single-output fuzzy inference systems designed from plant input-output
data using a neural network paradigm. This approach provides the control system with off-line learning capabilities to attain process optimization under changing operating conditions. The feedback control path is implemented as a PID-based decentralized (multiloop) control scheme with a loop interaction compensator. The compensator is equivalent to a disturbance feedforward compensator and is designed using the relative gain array technique. Both the control algorithms and the compensator are first order Sugeno-type fuzzy inference systems, scheduled in two dimensions (power and pressure) to achieve satisfactory disturbance rejection and uncertainty compensation during wide-range operation. The process operating window is partitioned to take into account the process nonlinear characteristics, and tuning is carried out by a genetic algorithm at the points of interest in the partitions.

The performance of the ICCS-MP is demonstrated through simulation experiments. Results show the feasibility of the proposed ICCS paradigm. An open purposeful self-governing overall unit control system for a FFPU can be systematically designed, built and upgraded to effectively satisfy arbitrary operation conditions. Remarkably, the ICCS paradigm provides a convenient conceptual framework such that the integration of applications can be carried out making use of the best characteristics that either algorithmic or heuristic techniques have to offer, while keeping large system complexity manageable.
TABLE OF CONTENTS

List of Figures xii
List of Tables xviii
List of Abbreviations xx
Acknowledgements xxii

Chapter 1 Antecedents and Generalities 1
1.1 Motivation 1
1.2 Background 4
  1.2.1 Overall unit control schemes 4
  1.2.2 Advanced control systems 9
  1.2.3 Intelligent hybrid approaches 13
1.3 Problem Statement 17
1.4 Objective and Scope 19
1.5 Thesis Overview 20
  1.5.1 Intelligent coordinated control system 21
  1.5.2 ICCS minimum prototype 23
1.6 Thesis Organization 26

Chapter 2 Power Unit Process Engineering and Models 27
2.1 Power Unit Process Engineering 27
  2.1.1 Need for a process engineering approach 28
  2.1.2 Performance upper limit of fossil fuel power units 29
  2.1.3 Exergy analysis of a fossil fuel power unit 36
  2.1.4 Overall operation of power units 42
  2.1.5 Qualitative dynamics of a power unit 43
2.2 Mathematical Models of Power Units 46
3.4 ICCS Minimum Prototype
   3.4.1 Functional scope of minimum prototype
   3.4.2 Physical scope of minimum prototype
   3.4.3 Control structure
   3.4.4 Feedforward/feedback strategy
   3.4.5 Feedforward requirements
3.5 Extended Coordinated Control
3.6 Summary

Chapter 4 Reference Governor
4.1 Introduction
4.2 Multiobjective Optimization
   4.2.1 Basic terminology
   4.2.2 General formulation of the multiobjective optimization problem
   4.2.3 MOOP solution methods
4.3 Goal Programming Method
   4.3.1 Goal programming basics
   4.3.2 Minsum goal programming formulation
   4.3.3 Optimization algorithm for FFPU
4.4 Supervisory Mapping Designer
   4.4.1 Feasibility regions of control signals
   4.4.2 Optimal steady-state control signals
   4.4.3 Calculation of set-points
   4.4.4 Unit responsiveness
4.5 Multiobjective Optimal Set-Point Mappings
   4.5.1 Specification of optimization goals
   4.5.2 Multiobjective optimal mappings
   4.5.3 Performance limit of process optimization
4.6 Summary
## Chapter 5  Feedforward Control Processor  
### 5.1 Introduction  
### 5.2 Neuro-Fuzzy Paradigm  
#### 5.2.1 TSK-type fuzzy systems  
#### 5.2.2 Feedforward neural-networks  
#### 5.2.3 Neuro-fuzzy paradigm  
### 5.3 ANFIS Methodology  
#### 5.3.1 Network representation of TSK-type fuzzy systems  
#### 5.3.2 Learning procedure of adaptive components  
### 5.4 Feedforward Control Processor  
#### 5.4.1 Neuro-fuzzy feedforward controller  
#### 5.4.2 Neural-network supervisory designer  
### 5.5 Design of Neuro-Fuzzy Controllers  
#### 5.5.1 Case studies  
#### 5.5.2 Effect of membership function number and training epochs  
#### 5.5.3 Feedforward fuzzy systems  
### 5.6 Summary

## Chapter 6  Feedback Control Processor  
### 6.1 Introduction  
### 6.2 Decoupled Multiloop Control  
#### 6.2.1 Decoupling control  
#### 6.2.2 Relative gain array  
#### 6.2.3 Compensator design  
#### 6.2.4 Decoupling feasibility and sensitivity  
### 6.3 Scheduled Fuzzy-PID Controller  
#### 6.3.1 Gain scheduling and multimode control  
#### 6.3.2 Fuzzy-PID controller  
#### 6.3.3 Application of the fuzzy-PID controller
6.4  Feedback Control Path Design
   6.4.1  2D scheduling of fuzzy-PID controllers
   6.4.2  2D Scheduling of interaction compensator

6.5  Summary

Chapter 7  Intelligent Coordinated Control

7.1  Introduction

7.2  Ancillary Applications
   7.2.1  Tuning of PID-based multiloop control system
   7.2.2  Genetic algorithm-based tuning
   7.2.3  Performance monitor
   7.2.4  Operating state monitor

7.3  Conventional Coordinated Control
   7.3.1  Control loop interaction and tuning
   7.3.2  Wide-range load tracking
   7.3.3  Changing operating point
   7.3.4  Process nonlinearity

7.4  System Partial Tests
   7.4.1  Feedforward control
   7.4.2  Set-point scheduler
   7.4.3  Interaction compensator
   7.4.4  Fuzzy-PID control

7.5  System Integral Tests
   7.5.1  Unit load demand step response
   7.5.2  Unit load demand ramp response
   7.5.3  Wide range cyclic operation
   7.5.4  Measurement noise
   7.5.5  Uncertainty on model parameters
   7.5.6  Variation of controller parameters
7.6 Comments 283

Chapter 8 Conclusions 284
8.1 Summary and conclusions 284
8.2 Contributions 286
8.3 Future research 288

Bibliography 291
LIST OF FIGURES

1.1 Coordinated boiler-follower control scheme 7
1.2 Coordinated turbine-follower control scheme 7
1.3 Generic coordinated control scheme 8
1.4 Intelligent coordinated control system block diagram 22
1.5 ICCS minimum prototype functional diagram 24
1.6 ICCS-MP control block diagram 25

2.1 Simple fossil-fuel power unit 29
2.2 Temperature-entropy diagram of simple ideal Rankine cycle 30
2.3 Superheat/reheat/regenerative power unit 33
2.4 Temperature-entropy diagram of Rankine cycle 34
2.5 Exergy audit at constant pressure 41
2.6 Exergy audit at sliding pressure 41
2.7 Open-loop response to step in steam valve position 45
2.8 Open-loop response to step in fuel valve position 45
2.9 Valve actuator block diagram 54
2.10 Power-pressure operating window 60

3.1 Context interaction of a power unit 81
3.2 Hierarchical control structure 83
3.3 Batch process control architecture 85
3.4 Intelligence functions in two-level system 87
3.5 Generic intelligence functions of ICCS 88
3.6 ICCS multiagent organization 90
3.7 Physical scope of ICCS 95
3.8 ICCS minimum prototype functional diagram 97
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.9</td>
<td>Physical scope of ICCS-MP</td>
<td>99</td>
</tr>
<tr>
<td>3.10</td>
<td>ICCS-MP core control structure</td>
<td>100</td>
</tr>
<tr>
<td>3.11</td>
<td>ICCS-MP control block diagram</td>
<td>101</td>
</tr>
<tr>
<td>3.12</td>
<td>Typical feedback control loop</td>
<td>102</td>
</tr>
<tr>
<td>3.13</td>
<td>Basic feedforward-feedback control loop</td>
<td>103</td>
</tr>
<tr>
<td>3.14</td>
<td>two-input two-output feedforward-feedback control</td>
<td>105</td>
</tr>
<tr>
<td>3.15</td>
<td>Equivalent feedforward-feedback control schemes</td>
<td>106</td>
</tr>
<tr>
<td>3.16</td>
<td>Conventional coordinated control</td>
<td>107</td>
</tr>
<tr>
<td>3.17</td>
<td>ICCS-MP as extended coordinated control</td>
<td>109</td>
</tr>
<tr>
<td>4.1</td>
<td>Reference governor in the ICCS-MP</td>
<td>114</td>
</tr>
<tr>
<td>4.2</td>
<td>Set-point scheduling in the reference governor</td>
<td>115</td>
</tr>
<tr>
<td>4.3</td>
<td>Three step set-point mapping design process</td>
<td>127</td>
</tr>
<tr>
<td>4.4</td>
<td>Fuel valve demand operating window</td>
<td>128</td>
</tr>
<tr>
<td>4.5</td>
<td>Steam valve demand operating window</td>
<td>128</td>
</tr>
<tr>
<td>4.6</td>
<td>Feedwater valve demand operating window</td>
<td>128</td>
</tr>
<tr>
<td>4.7</td>
<td>Multiobjective-optimal $E_{u1}-E_{d}$ relationships</td>
<td>137</td>
</tr>
<tr>
<td>4.8</td>
<td>Multiobjective-optimal $E_{u1}-P_{d}$ relationships</td>
<td>137</td>
</tr>
<tr>
<td>4.9</td>
<td>Multiobjective-optimal $E_{u1}-L_{d}$ relationships</td>
<td>137</td>
</tr>
<tr>
<td>4.10</td>
<td>Multiobjective optimal fuel control valve demands</td>
<td>138</td>
</tr>
<tr>
<td>4.11</td>
<td>Multiobjective optimal steam control valve demands</td>
<td>138</td>
</tr>
<tr>
<td>4.12</td>
<td>Multiobjective optimal feedwater control valve demands</td>
<td>138</td>
</tr>
<tr>
<td>4.13</td>
<td>Unit load demand cyclic pattern</td>
<td>139</td>
</tr>
<tr>
<td>4.14</td>
<td>Power set-point $E_{d}(t)$ trajectories</td>
<td>140</td>
</tr>
<tr>
<td>4.15</td>
<td>Power set-point $P_{d}(t)$ trajectories</td>
<td>140</td>
</tr>
<tr>
<td>4.16</td>
<td>Power set-point $L_{d}(t)$ trajectories</td>
<td>140</td>
</tr>
<tr>
<td>4.17</td>
<td>Fuel valve demand $u_1(t)$ trajectories</td>
<td>141</td>
</tr>
<tr>
<td>4.18</td>
<td>Fuel valve demand $u_2(t)$ trajectories</td>
<td>141</td>
</tr>
<tr>
<td>4.19</td>
<td>Fuel valve demand $u_3(t)$ trajectories</td>
<td>141</td>
</tr>
<tr>
<td>Section</td>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.3</td>
<td>Ideal decoupling</td>
<td>192</td>
</tr>
<tr>
<td>6.4</td>
<td>Simplified decoupling</td>
<td>193</td>
</tr>
<tr>
<td>6.5</td>
<td>Inverted decoupling</td>
<td>194</td>
</tr>
<tr>
<td>6.6</td>
<td>Inverse decoupling implementation of interaction compensator</td>
<td>203</td>
</tr>
<tr>
<td>6.7</td>
<td>Gain matrix condition number window</td>
<td>204</td>
</tr>
<tr>
<td>6.8</td>
<td>Gain matrix determinant value window</td>
<td>205</td>
</tr>
<tr>
<td>6.9</td>
<td>Gain scheduling control</td>
<td>206</td>
</tr>
<tr>
<td>6.10</td>
<td>Multi-mode control</td>
<td>208</td>
</tr>
<tr>
<td>6.11</td>
<td>Rule firing for any given operating condition</td>
<td>212</td>
</tr>
<tr>
<td>6.12</td>
<td>Fuzzy-PID controller</td>
<td>213</td>
</tr>
<tr>
<td>6.13</td>
<td>Actual system eigenvalue with largest variation</td>
<td>217</td>
</tr>
<tr>
<td>6.14</td>
<td>Operating space partitioning</td>
<td>217</td>
</tr>
<tr>
<td>6.15</td>
<td>Fuzzy sets of power scheduling variable</td>
<td>217</td>
</tr>
<tr>
<td>6.16</td>
<td>Relevant power-pressure characteristics in the operating window</td>
<td>219</td>
</tr>
<tr>
<td>6.17</td>
<td>Actual eigenvalues along power-pressure characteristics</td>
<td>220</td>
</tr>
<tr>
<td>6.18</td>
<td>Normalized eigenvalues along power-pressure characteristics</td>
<td>221</td>
</tr>
<tr>
<td>6.19</td>
<td>Partition of power range for upper-limit pressure</td>
<td>223</td>
</tr>
<tr>
<td>6.20</td>
<td>Pressure operating points for upper-limit pressure</td>
<td>223</td>
</tr>
<tr>
<td>6.21</td>
<td>Membership functions for upper-limit pressure</td>
<td>223</td>
</tr>
<tr>
<td>6.22</td>
<td>Partition of power range for constant pressure</td>
<td>224</td>
</tr>
<tr>
<td>6.23</td>
<td>Pressure operating points for constant pressure</td>
<td>224</td>
</tr>
<tr>
<td>6.24</td>
<td>Membership functions for constant pressure</td>
<td>224</td>
</tr>
<tr>
<td>6.25</td>
<td>Partition of power range for sliding pressure</td>
<td>225</td>
</tr>
<tr>
<td>6.26</td>
<td>Pressure operating points for sliding pressure</td>
<td>225</td>
</tr>
<tr>
<td>6.27</td>
<td>Membership functions for sliding pressure</td>
<td>225</td>
</tr>
<tr>
<td>6.28</td>
<td>Partition of power range for variable pressure</td>
<td>226</td>
</tr>
<tr>
<td>6.29</td>
<td>Pressure operating points for variable pressure</td>
<td>226</td>
</tr>
<tr>
<td>6.30</td>
<td>Membership functions for variable pressure</td>
<td>226</td>
</tr>
<tr>
<td>6.31</td>
<td>Partition of power range for lower-limit pressure</td>
<td>227</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6.32</td>
<td>Pressure operating points for lower-limit pressure</td>
<td>227</td>
</tr>
<tr>
<td>6.33</td>
<td>Membership functions for lower-limit pressure</td>
<td>227</td>
</tr>
<tr>
<td>6.34</td>
<td>Partitioning of power-pressure operating window (40% case)</td>
<td>228</td>
</tr>
<tr>
<td>6.35</td>
<td>Membership functions across the operating window (40% case)</td>
<td>229</td>
</tr>
<tr>
<td>6.36</td>
<td>Partitioning of power-pressure operating window (20% case)</td>
<td>230</td>
</tr>
<tr>
<td>6.37</td>
<td>Membership functions across the operating window (20% case)</td>
<td>230</td>
</tr>
<tr>
<td>6.38</td>
<td>Gain-scheduled interaction compensator</td>
<td>231</td>
</tr>
<tr>
<td>7.1</td>
<td>Response to power set-point step</td>
<td>249</td>
</tr>
<tr>
<td>7.2</td>
<td>Response to pressure set-point step</td>
<td>250</td>
</tr>
<tr>
<td>7.3</td>
<td>Response to level set-point step</td>
<td>251</td>
</tr>
<tr>
<td>7.4</td>
<td>Ramp load-tracking with step-tuned controller parameters</td>
<td>254</td>
</tr>
<tr>
<td>7.5</td>
<td>Ramp load-tracking with ramp-tuned controller parameters</td>
<td>255</td>
</tr>
<tr>
<td>7.6</td>
<td>Response to step in power set-point with ramp-tuned parameters</td>
<td>256</td>
</tr>
<tr>
<td>7.7</td>
<td>Response to step in pressure set-point with ramp-tuned parameters</td>
<td>257</td>
</tr>
<tr>
<td>7.8</td>
<td>Response to step in level set-point with ramp-tuned parameters</td>
<td>258</td>
</tr>
<tr>
<td>7.9</td>
<td>Responses to set-point steps at different operating point</td>
<td>260</td>
</tr>
<tr>
<td>7.10</td>
<td>Response to large set-point steps</td>
<td>261</td>
</tr>
<tr>
<td>7.11</td>
<td>Response to unit load demand ramp with feedforward control</td>
<td>263</td>
</tr>
<tr>
<td>7.12</td>
<td>Response to unit load demand ramp with 2 optimization objectives</td>
<td>265</td>
</tr>
<tr>
<td>7.13</td>
<td>Response to small set-point power step with interaction compensator</td>
<td>266</td>
</tr>
<tr>
<td>7.14</td>
<td>Response to small pressure set-point step with interaction compensator</td>
<td>267</td>
</tr>
<tr>
<td>7.15</td>
<td>Response to small level set-point step with interaction compensator</td>
<td>268</td>
</tr>
<tr>
<td>7.16</td>
<td>Responses to set-point steps at different operating point with fuzzy-PIDs</td>
<td>270</td>
</tr>
<tr>
<td>7.17</td>
<td>Response to unit load demand step with ICCS-MP</td>
<td>272</td>
</tr>
<tr>
<td>7.18</td>
<td>Response to fast large unit load demand ramp with ICCS-MP</td>
<td>275</td>
</tr>
<tr>
<td>7.19</td>
<td>Response to wide-range unit load demand cycle with ICCS-MP</td>
<td>276</td>
</tr>
<tr>
<td>7.20</td>
<td>Effect of measurement noise in wide-range cyclic response with ICCS-MP</td>
<td>278</td>
</tr>
<tr>
<td>7.21</td>
<td>Effect of control valve gain</td>
<td>280</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>7.22</td>
<td>Effect of control valve time constant</td>
<td>281</td>
</tr>
<tr>
<td>7.23</td>
<td>Effect of controller parameter variations</td>
<td>282</td>
</tr>
</tbody>
</table>
**LIST OF TABLES**

2.1 FFPU model variables 52  
2.2 FFPU model constant parameters 52  
2.3 FFPU rated operating conditions 53  
2.4 Actuator characteristics 54  
2.5 Operating points for input scaling 58  
2.6 Constant pressure equilibrium points 59  
2.7 Sliding pressure equilibrium points 59  
2.8 Upper pressure limit equilibrium points 61  
2.9 Lower pressure limit equilibrium points 61  

4.1 Parameters for optimization goals 135  
4.2 Cumulative values of objective functions 139  

5.1 Approximation accuracy of neuro-fuzzy controllers 177  
5.2 Fisul knowledge base for sliding-pressure: case with 3 mf 184  

6.1 Selected operating points 216  
6.2 Selected points for scheduling (40% case) 222  
6.3 Selected points for scheduling (20% case) 229  
6.4 Compensator elements for high-pressure limit 232  
6.5 Compensator elements for constant pressure 232  
6.6 Compensator elements for sliding pressure 233  
6.7 Compensator elements for variable pressure 233  
6.8 Compensator elements for low-pressure limit 234
7.1 Basic genetic algorithm 242
7.2 Parameters of PID controllers 252
7.3 Parameters of PID controllers for ramp response 259
7.4 Parameters of PID controllers for 1-objective case 269
7.5 Unit performance during cyclic operation 274
7.6 Unit performance during cyclic operation with noise 277
## LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>Artificial Neural-Network</td>
</tr>
<tr>
<td>BPLA</td>
<td>BackPropagation Learning Algorithm</td>
</tr>
<tr>
<td>BTG</td>
<td>Boiler Turbine Generator</td>
</tr>
<tr>
<td>CC</td>
<td>Coordinated Control</td>
</tr>
<tr>
<td>CIM</td>
<td>Computer Integrated Manufacturing</td>
</tr>
<tr>
<td>DMC</td>
<td>Dynamic Matrix Control</td>
</tr>
<tr>
<td>FBCP</td>
<td>FeedBack Control Processor</td>
</tr>
<tr>
<td>FFCP</td>
<td>FeedForward Control Processor</td>
</tr>
<tr>
<td>FFPU</td>
<td>Fossil Fuel Power Unit</td>
</tr>
<tr>
<td>FIS</td>
<td>Fuzzy Inference System</td>
</tr>
<tr>
<td>FRDV</td>
<td>Feasible Region in the Decision Variable</td>
</tr>
<tr>
<td>FROS</td>
<td>Feasible Region in the Objective Space</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GP</td>
<td>Goal Programming</td>
</tr>
<tr>
<td>IAE</td>
<td>Integral Absolute Error</td>
</tr>
<tr>
<td>ITAE</td>
<td>Integral Time Absolute Error</td>
</tr>
<tr>
<td>ISE</td>
<td>Integral Square Error</td>
</tr>
<tr>
<td>ITSE</td>
<td>Integral Time Absolute Error</td>
</tr>
<tr>
<td>ICCS</td>
<td>Intelligent Coordinated Control System</td>
</tr>
<tr>
<td>ICCS-MP</td>
<td>Minimum Prototype of the Intelligent Coordinated Control System</td>
</tr>
<tr>
<td>LSE</td>
<td>Least-Square Estimate</td>
</tr>
<tr>
<td>MAS</td>
<td>Multi-Agent System</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multi-Input-Multi-Output</td>
</tr>
<tr>
<td>MISO</td>
<td>Multi-Input-Single-Output</td>
</tr>
<tr>
<td>MOOP</td>
<td>Multi-Objective Optimization Problem</td>
</tr>
<tr>
<td>MPC</td>
<td>Multivariable Predictive Control</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>NFS</td>
<td>Neuro-Fuzzy System</td>
</tr>
<tr>
<td>NGP</td>
<td>Nonlinear Goal Programming</td>
</tr>
<tr>
<td>OCS</td>
<td>Optimal Contour Set</td>
</tr>
<tr>
<td>PI</td>
<td>Proportional Integral</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional Integral Derivative</td>
</tr>
<tr>
<td>RG</td>
<td>Reference Governor</td>
</tr>
<tr>
<td>SDV</td>
<td>Space of Decision Variables</td>
</tr>
<tr>
<td>SISO</td>
<td>Single-input-single-output</td>
</tr>
<tr>
<td>SOF</td>
<td>Space of Objective Functions</td>
</tr>
<tr>
<td>TSK</td>
<td>Takagi-Sugeno-Kang</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

I wish to thank Dr. Kwang Y. Lee, thesis advisor, for making the opportunity to come to Penn State available to me, and for encouraging me technically and spiritually during my doctoral studies. Also, I like to thank Dr. Mario Sznaier, Dr. Jeffrey S. Mayer, Dr. Robert M. Edwards, and Dr. Constantino M. Lagoa, for collaborating as doctoral committee members. To all of them for their valuable observations and suggestions, and more importantly for setting a high reference on the level of performance for a doctoral graduate, and for being a role model on how to achieve it.

In addition, I wish to thank the National Council of Science and Technology (Conacyt – Mexico), the Department of Electrical Engineering at the Pennsylvania State University, and the Electric Research Institute (IIE – Mexico) for their financial support, and the Fulbright Commission (IIE – USA) for its sponsorship, throughout my doctoral studies. Also, thanks to Mr. Marino Sanchez for trusting and backing me up.

I want to express my most profound gratitude to my beloved wife Sara for her encouragement and companionship, and for renouncing during all these years to her personal and professional aspirations, as a demonstration of love, so I can pursue this goal. I also want to express the greatest joy for being allowed to share these school years with my much-loved children Sara-Diana and Raul, whose presence and love kept my motivation atop. Finally, I am in eternal debt to my parents, Raul and Maricela, and to Mari and Charli, and Mine and Dani, for all their prayers and support back home.
CHAPTER 1

ANTECEDENTS AND GENERALITIES

In this chapter, some general facts and statements are made to establish the context of the present dissertation, and to provide a big picture of its products. In Section 1.1, the technological needs that motivate this dissertation are established. In Section 1.2, necessary background material on overall control of FFPUs is provided as the base from where this dissertation is launched. Section 1.3 is dedicated to specify the technological problem to be solved, and Section 1.4 establishes the objective and scope of the associated research activities. Finally, Sections 1.5 and 1.6 provide a preview of the final products and the contents of this dissertation, respectively.

1.1 Motivation

The current global scenario, regarding power generation, faced by electric utilities is characterized by many challenging problems. Among the most important are: aging of main equipment at power plants, financial investment uncertainty for new plant construction, competition among independent power producers to satisfy the energy demands of end users, and pressure to meet stringent government requirements to maximize the use of natural resources and to minimize environmental impact [Eden 1993, Dasgupta, et al. 1994, Merlin 1994].

Operation of Fossil Fuel Power Units (FFPUs), the most widely-used kind of units for power generation, has been heavily affected. Firstly, a FFPU must support the main objective of the power system, that is to meet the load demand for electric power at all times, at constant voltage and at constant frequency [Elgerd 1971]. Then, competition among utilities, and other market driven forces, have increased the usage of FFPUs in
load-following duties [Armor 1985]. Lately, stringent requirements on conservation and life extension of major equipment, and regulations on reduced environmental impact have to be fulfilled [Divakaruni and Touchton 1991]. In this regard:

- Cyclic operation practice demands the operation of FFPUs in a wide-range load-following fashion despite they were designed for constant load operation. Variable load demands may arrive from economic dispatch strategies evaluated at energy control centers or from system load upsets. Efficient and reliable load-following operation will assure effective dynamic participation in satisfying the daily, weekly and seasonally cyclic patterns of electric energy demand, and in following the unpredictable random system load upsets up to the unit physical or operative limitations.

- Plant life extension is important because it maximizes the use of capital assets, limiting down time and minimizing operational and maintenance costs. The main cause of duty life shortage for any system is high stress operation. In a FFPU, thermal stresses due to steam temperature and pressure fluctuations are particularly important. Most severe stress occurs during startup and sudden large load variations.

- A typical FFPU has a net efficiency between 30 and 35%, that is, heat rates between 11400 and 9800 Btu/KWh. Heat rate is affected by many factors, such as steam conditions, condenser pressure, cooling water temperature, ambient temperature, barometric pressure, etc. Heat rate increases when working at loads different from base load. Fuel consumption amounts and cost make heat rate improvement a critical issue of major economical impact.

- Unavoidable imperfections in the mixing of fuel and air make it necessary a certain excess of air to avoid incomplete fuel combustion, which will otherwise lead to black smoke and poisonous carbon monoxide production, and the risk of accumulating unburned fuel. Although necessary, excess air contributes to the formation of other undesirable emissions (sulfur dioxide, and nitrogen oxides), and decreases the boiler efficiency due to heat wasted in flue gases.
All above mentioned requirements, at least, call for the development of more comprehensive and flexible control methods [Nordbo, *et al.* 1992]. They should provide the necessary functions for high-quality wide-range load-following operation, and simultaneously, should satisfy the constraints in conservation and life extension of the main equipment, pollutant emissions, and fuel consumption, under changing physical and economical conditions. Consequently, even when load-tracking, and voltage and frequency stability have been the basic issues of concern, more effective control systems must also be designed to optimally satisfy arbitrary multiple, generally conflicting, operation objectives, so that a FFPU can successfully participate, under any operating scenario, in the everyday more demanding structure of a power system.

Additionally, under the current global trend toward market liberalization, an overall approach for operation and control of power units is of paramount importance for the survival of any electric utility [Armor and Weiss 1995]. When properly applied, plant-wide instrumentation and control systems can increase plant operating efficiency, operability and maneuverability, robustness and reliability, as well as plant availability, thus contributing to keep down fuel, operation, and maintenance costs, which account for most of the expenses in a power plant [Vasudeva 1991]. Therefore, there is an urgency to develop effective plant-wide automation systems, and consequently the associated overall unit control systems and strategies, to keep them running profitably.

Also, it should be noted that the intensive use of computer based instrumentation and control systems, with everyday more reliable and powerful general purpose information processing digital devices, allows system designers to focus more on the implementation of software applications to respond to the above mentioned challenges. Since software complexity, and the costs of its development and maintenance, could easily surpass those of the hardware in which it runs, great effort and care should be paid in the design and development of general and comprehensive software systems to ease the incorporation of advanced operation (*i.e.*, protection, control, and automation strategies) applications to enhance the performance of the power units [Garduno and Sanchez 1995, Garcia and Garduno 1998].
1.2 Background

It is the author’s thesis that a multidisciplinary approach is imperative for the design of an overall control system, and that this approach cannot be overemphasized when trying to produce a system with the best chances for being really useful. Within the bounds of this dissertation, the interaction among the areas of control engineering, process engineering, and software engineering is considered essential to create a feasible control system design for a FFPU. This section summarizes some relevant facts regarding overall control of FFUs. The discussion of facts on process engineering and software engineering is provided in Chapters 2 and 3, respectively.

Regarding power plant control, [Maffezoni 1996] highlights a contradiction between the current practice and the scientific literature proposing advanced control designs: “Control engineering practice provides improved solutions by gradual modifications of certain basic control structures, that appear to constitute still the core of the most recent commercial systems. On the contrary, the problem of control system structuring is often ignored in many papers considering new control designs based on multivariable techniques.” In what follows, both approaches, plus a third one that incorporates recent computational intelligence techniques, are discussed.

1.2.1 Unit control schemes

A FFPU supplies electric power as the result of a series of energy conversion processes. Roughly speaking, the main transformations are combustion of input fuel, steam generation, development of rotational motion, electric power production, and steam condensation. All these transformations compose a large thermodynamic cycle and are highly interdependent.

Current unit control strategies allow generation of the necessary power to satisfy the load demand while maintaining the balance among the transformation processes
within the unit. Mainly, they match the steam flow energy output of the boiler to the energy required by the turbine-generator to match the electric load at all times [Quazza and Ferrari 1972]. The unit control scheme constitutes the uppermost layer of the control system, and it is responsible for driving the boiler-turbine-generator set as a single entity. The dominant behavior of the unit is governed through the power and pressure control loops. Evolved from multiple single-input-single-output control loop (decentralized) configurations based on PID control algorithms, these strategies may be classified into three classes: boiler following (turbine leading) control, turbine following (boiler leading) control, and coordinated boiler-turbine control [Russell 1988].

Historically, boiler following schemes were the first to be used [Gery 1988]. In boiler following mode, the boiler awaits the actions of the turbine to match the requested generation. The turbine control valves regulate the steam flow into the turbine in terms of the power demand. Then, the boiler controls respond to the changes in steam flow and pressure. The throttle pressure deviation from the set point is used by the combustion control to regulate the fuel and air input to the furnace, and the steam production is modified to match the steam demanded by the turbine. The advantage of this approach is a fast response to load changes: the turbine being a fast acting device can respond very rapidly to load demands using the thermal energy stored in the boiler, while the boiler is then left to play catch-up. The disadvantage is that, in its pure form, this approach shows a less stable throttle pressure control since the boiler has a tendency to overshoot because it requires some time to match the turbine [Babcock and Wilcox 1978].

Turbine following mode began around the late 60’s and early 70’s [Gery 1988]. In the turbine following mode of control, the turbine follows the actions of the boiler to match the requested generation. The power demand is used by the combustion control at the boiler to adjust the fuel and air into the furnace to modify the steam production. Then the turbine controls respond by adjusting the throttle valves opening to keep the pressure at the set point value. The turbine is given full responsibility for controlling the throttle pressure. The advantage of this approach is its very stable response to load changes with minimal steam pressure and temperature fluctuations, since load changes depend on the
action of the boiler, which is a relatively slow device compared to the turbine. The main disadvantage is that this approach does not make use of the energy storage capability of the boiler, thus producing a rather slow response [Babcock and Wilcox 1978].

Originally proposed for once through units in the 50’s, coordinated control (CC) schemes were favored by the introduction of electro-hydraulic controls in the late 60’s [Gery 1988]. In CC mode the power demands are given simultaneously to the boiler and turbine controls. Depending on how the controlled and manipulated variables are paired, there are two possible modes for coordinated control: coordinated boiler-follower mode and coordinated turbine-follower mode [Landis and Wulfsohn 1988]. In coordinated boiler-follower mode, the load controller generates the demand to the steam throttle valve from the unit load demand and the measured generated power, and the pressure controller generates the demand to the fuel/air valves from the measured throttle pressure and the pressure set-point, which is obtained from the unit load demand through a non-linear mapping (Figure 1.1). In coordinated turbine-follower mode, the load controller generates the demand for the fuel/air valves from the unit load demand and the measured generated power. The demand to the throttle valve is calculated from the measured throttle steam pressure and the pressure set-point, which is obtained from the unit load demand through a non-linear characterization (Figure 1.2).

Coordinated control strategies are meant to synthesize the advantages of the other two strategies while minimizing their disadvantages, that is, they try to capture the stable response feature of the turbine following mode and the fast response characteristic of the boiler following mode. To attain fast response, the turbo-generator is allowed to draw upon the energy stored in the boiler. To achieve stability, the boiler control adjusts the fuel firing rate according to the required load, while keeping the turbine from exceeding the energy provided by the boiler. The unit response under CC is much faster than that of the turbine following scheme, but not as fast as the boiler following scheme [Babcock and Wilcox 1978]. Among the advantages of the coordinated control approach is the ability to easily implement rundown, runback, and variable pressure operation with precise load control, as required by supervisory economic load dispatch.
Figure 1.1 Coordinated boiler-follower control scheme.

Figure 1.2 Coordinated turbine-follower control scheme.
From a practical point of view, it would be highly desirable to have a general structure for coordinated control which can be either reconfigured to any of the modes of operation, or can be adjusted to show any intermediate degree of behavior between those modes. In [Tevera 1995] the reconfiguration alternative was explored. Since the proposed scheme is based on expert systems its description is provided in Section 1.2.3 about intelligent hybrid control approaches. Regarding the second alternative, an example of a generic CC structure that can accommodate any degree of control coordination was suggested in [IEEE 1991]. In this scheme (Figure 1.3), the load demand signal, developed either manually or by Automatic Generation Control, is first biased by the frequency error, to match the turbine governor droop characteristic, to form the desired MW signal for the unit. Then the actual measured MW is subtracted from the desired MW signal and biased by the pressure deviation to form an error which integrates into a change in the turbine load reference. The pressure error bias can be put through a dead band if desired, and the bias coefficient $K_p$ can also be made adaptive (i.e., proportional to the load

![Figure 1.3 Generic coordinated control scheme.](image-url)
The boiler control path can accommodate any degree of control action from complete boiler follow to complete turbine follow by appropriately setting the parameters in the various blocks \((K_{MP}, K_P, K_I)\). The model would also be capable of representing variable pressure control modes by manipulating \(K_{MP}\).

Despite the many implementations of CC by different manufacturers, current CC schemes basically consist of decentralized multiloop configurations of SISO controllers evaluating classical PI or PID algorithms. These schemes are seriously challenged by wide-range load-following operation requirements. In these conditions, performance may decrease due to the large nonlinear variations and coupling effects of process dynamics. Practical implementations exhibit complementary feedforward compensation schemes to enhance the basic multiloop coordinated control scheme by reducing the control loop interaction effects [Taft 1987]. Most realizations have evolved through various decades of research and practical experience, and most of them are confidential industrial property of their developers, and there is no general or widely accepted method to implement them; thus, more general and systematic design approaches are needed. A critical point is that the design of a mathematical-model-based interaction compensator might be unfeasible, intractable or become so complex that will preclude its application.

### 1.2.2 Advanced control systems

Although there is no general consensus on the definition of advanced control, many approaches that make use of mathematical models of the process, either at a design stage or during operation, are currently called advanced. An excellent review up to 1980 is presented in [Damborg, et al. 1980]. Some references that have transcended are mentioned in what follows for completeness.

Much attention has been paid to optimal control minimizing a quadratic performance index [McDonald and Kwatny 1973, Sandor and Williamson 1977]. The practical application of this technique for overall control has been limited by the
complexity of the implementation, sensibility effects due to model uncertainties, the need to include reset action in the controller, and the use of linear control techniques in a highly non-linear system. Even though not an overall unit control strategy, the most relevant application of the multivariable optimal control techniques is that of [Nakamura and Uchida 1989] which commissioned a linear quadratic regulator for the steam temperatures in the boiler of a 500 MW unit at the Kyushu Electric Company, Japan, in 1978. A breakthrough then, this kind of temperature control schemes, including other state-space identification and estimation, nonlinear mathematical programming, optimization and model reference adaptive control techniques, have become a standard feature in Japanese fossil power plants during the last 20 years, allowing them to operate with the highest levels of availability and thermal efficiency [EPRI 1985].

Research has also been done to apply decoupling methods to reduce or eliminate interaction effects between control loops. In [Chou, et al. 1977] a boiler-following unit control scheme is extended with a state feedback based compensator to reduce the system interaction effects of a 150 MW unit by Ontario Hydro in Canada. The compensator generates complementary signals for the four major control inputs as the weighted sum of the main control signals and the process state vector using a pair of constant gain matrices, which are calculated from a 9th order state-space linear model of the plant. The state vector is estimated using a Kalman filter. Simulation results showed improved load maneuvering capability in the 75% to 100% load operation range with reduced process variable errors and interaction effects. No details are given regarding the number of linear models used in the load range considered, neither on the way the gain matrices are switched in that same range. In [Borsi 1977] a decoupling controller based on a 12th order model was implemented in the real power plant. Compared to a traditional control, the approximately decoupled scheme yielded reduced control error, improved low load stability, simple parameter adjustments, and insensitivity to plant variations. The main difficulty with the decoupling approach was pointed out to be the design of the decoupling compensator. Design of dynamic compensators may be cumbersome, not always feasible, and their complexity may preclude their application.
In principle, adaptive controls seem to be particularly well suited for overall power plant control. The ability of the control system to deal with the changing dynamics of the process is very attractive for wide-range operation. In [Marc, et al. 1980] the effect of noise and parameter variation in the design of an adaptive controller for the steam generator at a 250 MW power plant is considered. In [Mabius, et al. 1980] a model reference adaptation mechanism is proposed to provide incremental control signals to be added to the control signals of an open loop control. Assuming operation around a fixed operating point, linear control theory is used to design the controller, and the control law parameters are adjusted using a Lyapunov-based scheme to assure stability of the closed loop. Unfortunately, only the design is presented with no results on the control system performance. Later, in [Fernandez-del-Busto, et al. 1984] an implicit model reference adaptive control algorithm is applied to the model of a 300 MW power plant. The adaptation algorithm is applied to each one of the main variables: steam pressure and temperature, reheat steam temperature, drum water level, and generated power. Comparison is made with a PID-based scheme showing improved results during load variations. Although not being an overall control scheme, adaptive control has been used successfully for main and reheat steam temperature control at some Chubu Electric Power units in Japan [Matsumura et. al. 1994]. The adaptive control works in parallel with the conventional PID boiler controllers. Plant dynamics are described with polynomials and their parameters are estimated from plant data. This approach provides reduced temperature excursions during load changes and has become a standard feature to achieve high thermal efficiency. Despite early great expectations adaptive overall control has not become an everyday reality. High dimension and nonlinearity of power units, particularly during wide-range variable pressure operation, still challenge adaptive methods.

Model-based predictive control [Camacho and Bordons 1999], in its many variations, is considered the most successful advanced control strategy for the petroleum and chemical industrial processes. A similar potential has been recognized for power plant control [Lectique, et al. 1978, Mehra, et al. 1982, Bender, et al. 1993]. In [Rossiter,
et al. 1991] the generalized predictive control algorithm (GPC) is applied to control the steam pressure and power output, and improved to deal with the slow boiler dynamics and the fast turbine dynamics. It is claimed, based on simulation studies, that the GPC algorithm provides a systematic approach to the boiler-turbine control problem, and provides improved performance over an optimally tuned PID-based control scheme. Although not an overall control strategy, the work of [Hogg and El-Rabaie 1991] applies a self-tuning multiloop generalized predictive control to regulate the superheater steam pressure, superheater steam temperature, and reheater steam temperature of a 200 MW plant model, over a wide range of operating conditions, showing a significant improvement over the conventional PI-based control strategy. In [Rovnak and Corlis 1991] the application of multivariable dynamic matrix control (DMC), based on linear models, is presented. No details are given on how the process non-linearities are handled by the linear auto-regressive sampled data model. Also, DMC is restricted to open-loop stable processes. In [Lu and Hogg 1995, Lu and Hogg 1997] both constrained and unconstrained model-based predictive coordinated schemes for power and pressure control are presented. Based on simulation experiments, it is claimed that the scheme allows optimal set-point response, rejection of known disturbance patterns, and provides enhanced robustness against model/plant mismatch under a wide range of operating conditions. Despite being a mature technique in both industry and academia, applications of MPC still have to show their value in real-world overall power plant control.

Robust control approaches have also been proposed for overall unit control. In [Weng and Ray 1997] a feedforward-feedback control strategy is proposed for wide-range robust load-tracking control of power plants. The feedforward control is intended to optimize the control inputs, taking into account their constraints, along wide operating ranges, and the feedback control is used to overcome plant modeling uncertainties and exogenous perturbations with guaranteed stability and performance robustness. The feedforward control solves a nonlinear programming problem for specified cost functional and constraints along a given load pattern. The feedback control is designed using the \( H_\infty \) singular perturbation technique. The proposed approach satisfies the
specified performance requirements in the 40%-100% load range and showed good rejection of anticipated disturbances. Some drawbacks are its extremely intensive computational power requirements, together with frequency response information and a linear model of the plant.

Despite the great advance in control system theory, real world implementations of advanced schemes for FFPU are still scarce. To a great extent this is due to the fact that the proposed schemes do not perform well under real world requirements such as real-time performance, wide range operating capacity, high dimension and complex process dynamics, or they assume conditions that are rarely given in practice such as Gaussian noise, constant process parameters, correct and exact measurements, well known process dynamics, etc. Perhaps, the main disadvantage of most advanced methods is the requirement of precise mathematical models for design and during operation. High dimension and complexity of power units have precluded the implementation of full scope mathematical model based centralized control schemes. On the other hand, this situation has motivated the prolonged existence of multiloop configurations based on conventional PID control algorithms.

1.2.3 Intelligent hybrid approaches

There is still a third approach that applies the recently emerged artificial intelligence techniques, or a mixture of them with conventional or advanced controls, in an attempt to deal with complex issues that have not been solved satisfactorily by the previously discussed techniques.

In [Mamdani 1974, Mamdani and Assilian 1975] a controller based on fuzzy set theory [Zadeh 1965, Zadeh 1968, Zadeh 1973] is introduced for the first time. A fuzzy algorithm, that emulates the reasoning procedure of a human operator, is used to control a laboratory-built boiler steam engine. Steam pressure and rotor speed are regulated by manipulating the boiler heat input and the engine throttle opening according to a set of
linguistic procedural rules. Experiments showed similar or better results than classical controllers. This work demonstrated the feasibility to easily build effective real-time decision algorithms, inaugurating a new era in control engineering.

In [Ray and Majumder 1985] nonlinear decoupling theory and fuzzy logic are used to regulate the drum pressure and the drum water level in a 200 MW power unit at different operating conditions. First, a decoupling control law is designed for a third order nonlinear plant model, under the assumption that the system parameters are known exactly, causing it to appear as a set of non-interacting single-input-single-output subsystems. Then, fuzzy logic controllers are used for closed-loop control in each subsystem. Good performance of the control scheme is obtained when evaluated with respect to state perturbations, piecewise and random parameter variations, set-point transfer, parameter sensitivity, improper decoupling, and lack of decoupling.

In [Marcelle, et al. 1994] a control scheme that combines fuzzy logic and optimal control for cyclic operation of large steam turbines is presented. Cumulative turbine stress and load tracking error are minimized by coordinating the control of the turbine governor valve and the boiler steam pressure. A fuzzy system prioritizes the conflicting performance objectives and provides the weights in a cost function to be used by the optimal controller. Then, a model predictive control algorithm is used to calculate the optimal valve and pressure set-point values for tradeoff between good load tracking and stress minimization. Better performance was obtained when compared against conventional constant pressure and variable pressure controller, and also against a model predictive controller with fixed gains and fixed prediction horizon.

In [Dimeo and Lee 1995] a genetic algorithm is used to optimally tune a PI-based coupled control scheme and a multivariable optimal state feedback controller for a third order nonlinear power unit model. The control objective is to track step demands in power output and steam pressure. While the application feasibility of the genetic algorithm as a tuning mechanism is proved correctly, there is some unfortunate misconceptions in the proposed control structures. The location of the cross-coupling gains in the PI-based coupled control scheme is not convenient, since it promotes
oscillatory response, and the optimal control scheme mistakenly assumes that all state variables are available for simulation.

In [Tevera 1995] a reconfigurable unit control system that can be switched among different control strategies (i.e., boiler follow, turbine follow and coordinated control) is presented. System reconfiguration is carried out on-line by four expert systems working cooperatively at the supervisory level. The first one identifies the operation state and requirements for the unit. The second evaluates the performance of the control scheme in operation. The third one evaluates the performance of the other schemes under the current conditions. Finally, the fourth makes a decision on which scheme is the most convenient and switches the scheme if allowed by the operator. The control was implemented successfully in a system for control software development and evaluation [Garduno, et al. 1990] using a full-scope power unit model.

In [Ben-Abdennour 1996, Ben-Abdennour and Lee 1996] a two-level hierarchical control system is developed to increase the degree of autonomy of the controlled power plant. In the lower level, the local boiler and turbine controllers, arranged in a boiler-following-turbine CC configuration, are implemented using the LQG/LTR technique. The generator is controlled independently by a standard excitation system. In the upper level a set of fuzzy systems, using process information provided by a Kalman filter, perform fault diagnosis and fault accommodation functions and supply corrections to the set-points for the local controllers. Simulation results show substantial improvements for all single fault and most of the multiple fault study cases.

In [Kallapa, et al. 1997, Kallapa 1998] a two-level hierarchical control system for life extension of power plants that takes into account fatigue and structural damage in the steam headers and the steam generator tubes is proposed. A feedforward-feedback scheme is used at the direct control level. The feedforward control signal is obtained from the optimization of a weighted sum of the measures of plant performance and accumulated structural damage. Optimization is performed on-line or off-line when the operation strategy (unit load demand) is known a-priori or a-posteriori, respectively. A set of robust $L_2$ feedback controllers is synthesized based on linear plant models, which
are then gain-scheduled to achieve wide-range operation. The supervisor is a fuzzy expert system that schedules and generates the reference signals for the feedback control. Simulation experiments showed good performance and enhanced component life extension.

In [Garduno and Lee 1997] A two-level hierarchical control scheme for wide-range operation of fossil fuel power units is presented. At the supervisory level, a fuzzy reference governor generates, according to a variable pressure operating policy, the set-point trajectories to command the unit along any load demand pattern. At the control level, a feedforward-feedback control strategy is implemented. The feedforward control path contains a set of multi-input single-output fuzzy inference systems, designed from steady-state input-output plant data. The feedback control path consists of PID controllers in a multi-loop configuration, as currently available at power units. With this strategy, the feedforward path provides most of the control signal for wide-range operation, diminishing the control effort on the PID controllers. The feedback path supplies the complementary control signal component for regulation and disturbance rejection in small neighborhoods around the commanded trajectories. Simulation results demonstrate the feasibility of the control scheme to attain cyclic load-following operation.

Large research initiatives have been undertaken in Japan to develop hybrid schemes to obtain technological advantage in power plant operation [Matsuoka, et al. 1993]. After the epoch-making implementations of multivariable optimal control, which have been in operation during the last twenty years, the Japanese are producing a second wave of technological breakthroughs in power plant control through the implementation of hybrid schemes based on both algorithmic formulations and neuro-fuzzy systems. More complete solutions are seek to the challenges posed by: 1) larger and more complex processes, including their mathematical modeling, 2) stronger nonlinearities, time varying couplings and interaction among subsystems, and 3) satisfaction of tighter requirements in system stabilization, coordination, and optimization.

In the USA, the Intelligent Control Systems Initiative, launched jointly by the Electric Power Research Institute (EPRI) and the National Science Foundation (NSF) in
1993, consisted of twenty one research projects aimed at finding innovative and valuable applications and at establishing fundamental theory or reliable procedures for the design and development of adaptive and intelligent systems [Wildberg 1995]. Some approaches to intelligent control under investigation are: 1) supervisory planning off-line and on-line by automated reasoning, or learned from human examples, 2) qualitative modeling using expert systems and fuzzy logic, and 3) computational intelligence and machine learning using neural networks, genetic algorithms, and evolutionary or reinforcement learning methods, sometimes combined with mathematical optimization techniques. In [Lee 1997] a summary of some of the research activities on intelligent distributed control of power plants at the Power Systems Control Laboratory of the Pennsylvania State University is presented.

### 1.3 Problem Statement

The author’s leading vision for the development of control systems for power units is that of the development of artificial energy management systems designed after the regulatory processes of the highly energy-efficient living beings, where a power unit will play the role of a self-sustained cell-like system for local energy transformation and production, within a highly integrated and coordinated macro-organism (i.e., the earth’s global power system) intelligently controlled by a truly brain-like computer system. Ideal for now, this vision demands substantial changes not only on control systems technology, but also on power systems technology, at least. Nevertheless, some advance may be done by formulating the problems to be solved, investigating options for their solution, and pointing out new research needs. In this sense, the basic nature of this research project constitutes an exploratory journey in the mentioned direction.

After having identified the current global operation scenario faced by FFPUs, reviewed and classified the most relevant propositions on overall unit control in the technical literature, and described the long-range leading vision for power unit operation,
a technology innovation process can be formulated. To this aim, a power plant is considered as a malleable component available for direct control, that can be seen, globally, as a single actuator, and locally, as a large-scale complex nonlinear system, where top priority is given to promote the highest levels of automation in power systems through the development of an overall unit control system, with a clear mechanism to deal systematically with the ever changing, present and future, operation requirements for a power unit. Therefore, the main requirements for the control system are to posses:

- Open design to accommodate diverse functional and operational requirements to achieve versatility and flexibility, for application in a highly variable market-driven environment.
- Simple end-user realization and high performance, clearly linked to proven current automation and control practice and technology.
- Structured system architecture, that will eventually lead to the functional specification and development of the corresponding automation and control software.

Consequently, the two ends of the technology innovation process are fixed as follows. First, coordinated control strategies constitute the uppermost control level in current FFPUs and so, they are responsible for driving the boiler-turbine-generator set as a single entity. Second, a FFPU is envisioned as a complex process, subject to multiple changing operation conditions, that should perform as an intelligent system, for which an advanced integrated control concept is needed.

The present stage of technological development suggests to design a power unit control system to support and to facilitate the implementation of large scale hierarchical and decentralized power control systems, and to incorporate artificial intelligence techniques to deal systematically with the complexity of the process and its demanding operation requirements. Therefore, in this research project a generalized unit control concept that extends the capabilities of current coordinated control schemes is contributed. This concept is presented as the Intelligent Coordinated Control System (ICCS), which establishes a general frame for the development of overall control schemes. In general, the ICCS extends the scope of current coordinated control schemes
to attain wide-range cyclic operation with optimization of multiple operation requirements (i.e., load tracking, duty life, heat rate, and pollutant emissions). To do so, the ICCS incorporates functions of machine learning, behavior monitoring, adaptation mechanisms, command generation, and control evaluation, all of them needed to exhibit effective and versatile self-governing operation characteristics.

1.4 Objective and Scope

The main objective of this research project is to contribute to increase the automation level of power plants through the extension of the coordinated control concept at fossil-fuel power units and its realization via artificial intelligence techniques.

To keep this research project down to earth, the following points establish the scope of it:

- All research and development in this project will be focused on conventional drum-type Fossil-Fuel Power Units (FFPU), since they represent the most commonly used technology to generate electric power in bulk quantities, and constitute the main way to regulate the most important parameters (power, frequency and voltage) affecting the electric energy quality in interconnected power systems.

- Only normal dynamic operation of a FFPU will be considered. The project concentrates in the control system needed in a normal operation state, which accounts for 99% of the operating time. No explicit attempt will be made to support the alert, emergency, extreme, and restorative states, for which proper strategies should be developed to complement the normal state control strategy. It is worth mentioning here that normal state does not mean steady state, since in reality a generating unit never operates in true steady state.

- The control system design will comprehend only the fundamental control strategy. The development of the automation, interlocking, and protection strategies, as given in any practical control systems will not be undertaken. Also, by fundamental we
refer to the core control functions employed to drive the process during the load operating phase, neither start-up or shut-down functions will be considered.

- The development of the control system will be limited to the system proposition, and its feasibility demonstration. Proposition accounts for the conceptual model development, and feasibility demonstration for prototype implementation. All prototypes will be up to the point of simulation in a personal computer or workstation platform, in a software laboratory environment. No attempt will be made to implement the system in other platform with demonstrative purposes nor a real-world application will be undertaken. This is in accordance with the basic research nature of this project.

- The conceptual model development assumes current power generation technology. No proposition will be made toward the use of different or newer technology (i.e., ultra-supercritical plants, sliding pressure units, fluidized bed combustion, etc.) in the power generation process. Also, this project concentrates on the functional aspects of the control system not in the hardware technology being used for its implementation.

- The prototype development comprehends a minimum set of functions necessary to promote total power system automation and regulation, offering maximum flexibility to support load control under market deregulation and multi-unit control in power plants schemes, which are still awaiting for definition and standardization.

- The proposed control approach will be evaluated and compared to conventional approaches. Even if the kind of control system being proposed does not currently exist, evaluation will be made with respect to load-following performance, which is the most important requirement to be satisfied.

1.5 Thesis Overview

As results of the technology innovation research project undertook in this dissertation, the following two main products were obtained:
• The conceptual model of a generalized unit control structure for fossil fuel power units, named the Intelligent Coordinated Control System (ICCS).
• A minimum prototype of the ICCS (ICCS-MP) in the form of an intelligent hybrid control system.

The ICCS is a general functional framework for the development of full scope power unit control systems that satisfies the requirements listed in Section 1.3, whereas the ICCS-MP is a realization that demonstrates the feasibility of the ICCS.

1.5.1 Intelligent coordinated control system

The definition and functional specification of the ICCS is synthesized as an extension to the batch control system structure proposed by [Rosenof and Ghosh 1987], taking into account the two most relevant propositions for the implementation of intelligent machines and systems [Saridis and Valavanis 1988, Albus 1991], and the trend toward computer integrated processes [Williams 1989]. It is also a continuation of the author’s own effort for control system technology development in power plants [Garduno, et al. 1990, Garduno, et al. 1996] and the supervision of the work by [Tevera 1995, Mundo 1995, Ramirez 1996, and Garcia 1997] for the development of control systems to increase the level of automation in power plants.

Basically, the ICCS can be seen as a frame in which control functions already performed by separate controllers, and new knowledge-processing functions not yet available at FFPUs, can be incorporated in a harmonious and unified way to define an overall unit control strategy.

From a broader perspective, the ICCS is conceived as a generalized unit control scheme to be implemented as an open multiple-agent intelligent hybrid system, which, based on a structured and comprehensive functional specification, can accommodate the functions needed for current and anticipated changing operation requirements for FFPUs within highly automated power systems.
The actual ICCS architecture resembles a two-level hierarchical system (Figure 1.4), where the upper level, mainly knowledge-driven, performs the supervisory functions needed to provide self-governing operation characteristics, while the lower level, mainly data-driven, performs the fast reactive behavior functions necessary for real-time control and protection.

Figure 1.4 Intelligent coordinated control system block diagram.
The high level knowledge-processing functions are intended to provide the intelligent behavior necessary to attain wide-range operation under changing operation scenarios. These functions provide the means to deal with the nonlinear strongly coupled multivariable process dynamics, external weather (i.e., atmospheric pressure, temperature and humidity) and physical perturbations (i.e., unexpected large load variations), process optimization under multiple conflicting objectives and operation-knowledge uncertainty, and system performance quantification. These high-level functions comprehend behavior monitoring, machine learning and adaptation mechanisms, optimization and command generation, and control evaluation, all of which are needed to exhibit effective and versatile self-governing operation characteristics.

The low-level data-driven functions are intended to provide the fast reactive behavior functions necessary for real-time control and protection evaluation. These functions comprehend sequence control, regulatory control, and protection and interlocks. The execution managers, inference mechanisms, communication, data acquisition, and operation interface functions are considered as support functions for the ICCS.

1.5.2 ICCS minimum prototype

Recognizing that the development of the ICCS for a real-world power plant is a formidable task, only the development of a minimum prototype (ICCS-MP) is attempted in this research project. The ICCS-MP comprehends a minimum set of functions that preserves the essential objective of the ICCS. First, the ICCS-MP adopts the structure of the ICCS, so that participation in total power system automation is guaranteed. Second, realizing that most of the time a FFPU is in the normal operating state, where any minimum performance improvement will certainly yield enormous benefits, major priority of the ICCS-MP is to provide the means to achieve optimized wide-range cyclic operation, by being able to follow any given unit load demand profile issued by upper
level economic dispatch and unit commitment agents, and optimally accommodate an arbitrary number of generally conflicting operation objectives.

Developed through several stages, the ICCS-MP finally implements an intelligent hybrid multi-agent supervisor with evaluation of regulatory control in the direct control level. The supervisory functions include optimization and command generation, learning and adaptation, and performance and state monitoring (Figure 1.5). The direct level consist of a multivariable feedforward/feedback control scheme. The implementation core of the system is formed by three modules: set point scheduler, feedforward control processor, and feedback control processor (Figure 1.6). The performance and state monitoring, and learning and adaptation functions are executed either under demand in an off-line basis or are implicitly included in the main modules.

![Figure 1.5 ICCS minimum prototype functional diagram.](image-url)
Given an arbitrary unit load demand trajectory, the set-point scheduler generates set-point trajectories for the lower level control loops through a group of optimally designed mappings. The set-point mappings are designed by solving a multiobjective optimization problem, for which the objective functions and their preferences can be set arbitrarily. This approach allows for process optimization, and provides the means to specify the operating policy to accommodate a great diversity of operating scenarios characterized by multiple operating objectives. The proposed two-degrees-of-freedom feedforward-feedback control scheme is an extension of the general linear single-input-single-output feedback control scheme, with both reference feedforward and disturbance feedforward control actions, to the nonlinear multivariable case to achieve wide-range operation. The feedforward control processor is implemented using a set of multi-input-single-output fuzzy inference systems, designed from steady-state plant input-output data, using a neural network learning technique. The feedback control path is implemented as a decentralized (multiloop) control scheme based on fuzzy-PID controllers and a loop interaction compensator. The compensator is equivalent to a disturbance feedforward compensator and is designed as an inverse decoupler using the relative gain array technique. The fuzzy-PID controllers amalgamate the gain-scheduling and multimode control techniques using a Sugeno-type fuzzy inference systems. Both the fuzzy-PID control algorithms and the compensator are scheduled in two dimensions (power and pressure) to achieve satisfactory wide-range operation.
1.6 Thesis Organization

This dissertation is organized in eight chapters. Chapter 1 motivates and states the research problem from the control engineering point of view, and overviews the solution as the design of a generalized overall unit control system for power units. Chapter 2 presents some fundamental process engineering facts used to define the control objectives and operating policies of a fossil-fuel power unit. Also, the mathematical model of a power unit is presented in several versions. Chapter 3 presents a detailed description of what the complete ICCS should be. The generic structure of the ICCS is described from various dimensions. Also, describes the structure of the ICCS-MP. Chapters 4, 5, and 6 provide a detailed description of the ICCS-MP main components: reference governor, feedforward control processor, and feedback control processor, respectively. Each description includes justification, theoretical foundation, design, and analysis. Chapter 7 presents the ICCS-MP fully integrated, and shows its performance based on simulation experiments. Finally, Chapter 8 presents a summary of the dissertation, its contributions, conclusions, and makes some suggestions for future research.

Alternatively, the organization of this dissertation can be viewed from a software engineering perspective by relating the development of the ICCS to a software process life-cycle. Chapters 1 and 2 constitute the specification of requirements. Chapter 3 constitutes the design stage. Describes the conceptual model of the ICCS as a multiagent system, and specifies a minimum prototype (ICCS-MP) for realization. Chapters 4, 5, and 6 undertake the implementation and validation (module testing) of the ICCS-MP main components. Finally, Chapter 7 describes the integration and validation (system testing) of the ICCS-MP.
CHAPTER 2

POWER UNIT PROCESS ENGINEERING AND MODELS

This chapter focuses on relevant process engineering concepts and various models of a fossil fuel power unit (FFPU), which constitute the foundation for the design of an overall control strategy in the following chapters. The process engineering concepts provide a basic understanding and insight into the process, which cannot otherwise be gained solely by direct inspection and analysis of the model mathematical equations. In fact, they must be known before analyzing any dynamic model, so they are provided in Section 2.1. In Section 2.2, the characteristics of the mathematical model as required in this dissertation are discussed. Fortunately, a very convenient model already exists for wide-range simulation of the essential power unit dynamic effects, and the design of overall unit control systems. In Section 2.3, the selected nonlinear mathematical model is described, and in Sections 2.4, and 2.5, it is formulated in static (direct and inverse), and linear (state-space and transfer matrix) versions, respectively. Finally, Section 2.6 briefly summarizes and comments this chapter.

2.1 Power Unit Process Engineering

As mentioned in Chapter 1, the multidisciplinary approach amalgamating control engineering, process engineering, and software engineering concepts, for the design of the overall power unit control system is a central issue in this dissertation. In this chapter, the process engineering aspects are highlighted by noticing that enhanced performance of a fossil fuel power unit can only be achieved when the control system operates in harmony with basic thermodynamic engineering facts and well established operation practices. Perhaps the most relevant issue in the interdependence between process and
control engineering is the definition of control objectives and process operation policies, congruent with each other. In this regard, the minimum process engineering concepts that must be taken into account for overall process optimization under arbitrary operation scenarios are presented.

2.1.1 Need for a process engineering approach

The rationale behind the integration of process engineering and control engineering is straightforward, albeit not always honored. It is amazing to see how many published control strategies ignore basic simple operation facts and needs, thus limiting their application beyond computer simulations (no explicit references made for politeness). It is not just the mathematical model of the process that should be considered, but also the expected operation dialogs for proper automation and basic thermodynamic principles. It is not rare to find optimal controllers making no sense and blind optimizations against the intended purpose of the given plant, thus reflecting lack of knowledge of the real process in the race to apply the latest control techniques to get published. Or even worse, there is tons of designs of panacea-like control strategies valid and successful for the very many different operation modes or operating states of a process, or strategies that mix different functions (i.e., protection, control, supervision, etc.) without any regard to priorities or preferences in the control actions, thus giving the clue that the designers have no idea of what the big picture of the control system is.

In what follows, a look is taken to some facts that should be taken into account whenever a control system for power plants is designed, as well as when the overall system operation and control goals are established. This minimum set of concepts, as applied to this dissertation, include basic knowledge about the upper limits in power plant performance, the assessment of real losses in a power plant, basic overall plant operation procedures, and a qualitative notion of overall plant dynamics. These concepts will be assumed implicitly in the coming chapters, in addition to the various plant models.
2.1.2 Performance upper limit of fossil fuel power units

A FFPU produces electric power from fossil fuel through several energy conversion processes, using water as a working fluid. The chemical energy of the fossil fuel is transformed into steam thermal energy by the boiler, then it is transformed into rotational mechanical energy by the turbine, and finally it is transformed into electric energy by the generator (Figure 2.1). Concurrently, the working fluid is alternately vaporized and condensed in a closed circuit following a thermodynamic cycle.

Since the efficiency of electric generators is very high (~98%), the overall efficiency of the power unit is determined by the performance of the remaining components, mostly by the characteristics of the thermodynamic cycle of the working fluid, which is examined in this section. In the ideal steady state operation case, heat transfer with the surroundings is neglected, potential and kinetic energy changes are
ignored, and internally reversible processes are assumed, to simplify the analysis. Under these conditions, the transit of the working fluid through the boiler and condenser is isobaric (constant pressure), while through the turbine and pump is isentropic (constant entropy). The corresponding temperature-entropy diagram of an ideal simple Rankine cycle is shown in Figure 2.2, where states 1 through 4 correspond with points 1 through 4 in Figure 2.1.

![Temperature-entropy diagram of simple ideal Rankine cycle.](image)

From point 1 to 2, liquid water undergoes an isentropic compression in the pump, rising its pressure significantly. From point 2 to 3, the heat released by fuel combustion is transferred to the working fluid in the boiler, as it is converted from liquid water to saturated vapor at constant pressure. The amount of heat transferred to the working fluid to increase its temperature from state 2 to state \( c \) is called sensible heat. The temperature at state \( c \), at which the water starts boiling is known as saturation temperature, and the amount of heat added from state \( c \) to point 3, without temperature increase, is named latent heat. From point 3 to 4, the saturated vapor undergoes an isentropic expansion in
the turbine, decreasing its pressure significantly to the value at the condenser. Vapor expansion is used to produce the turbine shaft rotational motion to propel the electric generator. Finally, from point 4 to 1, heat is transferred from the working fluid to the cooling water in the condenser as it is converted from saturated vapor to liquid water at constant pressure.

The principal work, heat transfers, and thermal efficiency of the plant may be accounted in a coarse but insightful way using the same assumptions as above and the simple plant in Figure 2.1. Using the principles of mass and energy conservation, expressions for the rate of change of energy or work can be obtained for a unit of mass of working fluid following the cycle path. Taking a control volume around the feedwater pump, from point 1 to 2, yields:

\[
\frac{\dot{W}_p}{\dot{m}} = h_2 - h_1
\]  

(2.1)

where \(\dot{W}_p / \dot{m}\) is the rate of power input per unit of mass of steam passing through the pump, while \(h_1\) and \(h_2\) are the water enthalpy at point 1 and 2 respectively. Taking a control volume enclosing the boiler pipes and drum, from point 2 to 3, yields:

\[
\frac{\dot{Q}_{in}}{\dot{m}} = h_3 - h_2
\]  

(2.2)

where \(\dot{Q}_{in} / \dot{m}\) is the rate of heat transfer from the energy source into the working fluid per unit mass passing through the boiler, and \(h_3\) is the water enthalpy at point 3. For a control volume around the turbine, from point 3 to 4, gives:

\[
\frac{\dot{W}_t}{\dot{m}} = h_3 - h_4
\]  

(2.3)

where \(\dot{W}_t / \dot{m}\) is the rate at which work is developed per unit mass of steam through the turbine, and \(h_4\) is the water enthalpy at point 4. Finally, considering a control volume enclosing the condensing side of the heat exchanger at the condenser, from point 4 to 1:

\[
\frac{\dot{Q}_{out}}{\dot{m}} = h_4 - h_1
\]  

(2.4)

where \(\dot{Q}_{out} / \dot{m}\) is the rate at which energy is transferred by heat from the working fluid to the cooling water per unit mass of working fluid passing through the condenser. Then, the
thermal efficiency of the plant is given by the ratio from the net work output to the fuel input heat energy:

\[ \eta = \frac{\dot{W}_n / \dot{m} - \dot{W}_p / \dot{m}}{Q_{in} / \dot{m}} = \frac{(h_3 - h_4) - (h_2 - h_1)}{h_3 - h_2} \]  

(2.5)

Taking into account that the net work output equals the net heat input, the thermal efficiency can also be written as:

\[ \eta = \frac{\dot{Q}_{in} / \dot{m} - \dot{Q}_{out} / \dot{m}}{Q_{in} / \dot{m}} = 1 - \frac{\dot{Q}_{out} / \dot{m}}{Q_{in} / \dot{m}} = 1 - \frac{h_4 - h_1}{h_3 - h_2} \]  

(2.6)

Normally, the boiler and condenser pressures are known, as well as the net power output. From these values, steam property tables, and considering the simplifying assumptions, it is possible to calculate the water enthalpies, the thermal efficiency, the steam mass flow rate, the rate of heat transfer to the working fluid at the boiler, and the rate of heat transfer from the working fluid at the condenser, thus having a fairly good characterization of the power plant at steady state conditions.

A more intuitive expression for the thermal efficiency may be obtained in terms of average temperatures during the heat interaction processes. The total area \( a-b-1-2-c-3-4-a \) represents the heat transfer into the working fluid per unit mass through the boiler:

\[ \frac{\dot{Q}_{in}}{\dot{m}} = \int_{2}^{3} T \, ds = T_{in} (s_3 - s_2) \]  

(2.7)

where \( T_{in} \) is the average temperature from state 2 to 3, and \( s_2 \) and \( s_3 \) are the water entropy at points 2 and 3, respectively. Similarly, the area \( a-b-1-4-a \) is the heat transfer from the condensing steam per unit mass through the condenser:

\[ \frac{\dot{Q}_{out}}{\dot{m}} = \int_{1}^{4} T \, ds = T_{out} (s_4 - s_1) = T_{out} (s_3 - s_2) \]  

(2.8)

where \( T_{out} \) is the temperature on the steam side of the condenser, and \( s_1 \) and \( s_4 \) are the water entropy at points 1 and 4, respectively. Then, by substituting (2.7) and (2.8) in (2.6) the thermal efficiency can be written as:

\[ \eta = 1 - \frac{\dot{Q}_{out} / \dot{m}}{\dot{Q}_{in} / \dot{m}} = 1 - \frac{T_{out}}{T_{in}} \]  

(2.9)
This very important relationship provides a major guideline to improve plant performance through plant design. The thermal efficiency will increase as the average input temperature \( T_{in} \), at which energy is added by heat transfer, increases and/or as the temperature \( T_{out} \) at which energy is rejected decreases. This is equivalent to shift section 2-3 upwards, and/or section 4-1 downwards, in the simple Rankine cycle of Figure 2.2. In turn, this can be physically accomplished by operating the power unit at higher boiler temperatures and pressures to raise points 2 and 3, and at lower condenser temperatures and pressures to lower points 1 and 4.

In actual power plants, various modifications are incorporated to improve the overall thermal efficiency by increasing the average input temperature of heat addition to the working fluid. Main methods include main steam superheating and reheating, and feedwater regenerative heating, as shown in Figure 2.3. The corresponding changes in the thermodynamic cycle are shown in Figure 2.4. In what follows, “point” refers to a physical location in Figure 2.3, and “state” refers to a thermodynamic state in Figure 2.4.

Figure 2.3  Superheat/reheat/regenerative power unit.

\( B \) = boiler, \( G \) = governor, \( cp \) = condenser pump, \( fp \) = feedwater pump
\( lph \) = low-pressure heater, \( hph \) = high-pressure heater, \( da \) = deareator (open heater)
\( ec \) = economizer, \( ww \) = waterwall, \( sh \) = superheater, \( rh \) = reheater
Feedwater regenerative heating is carried out by multiple feedwater heaters, usually of three different kinds: closed heaters, dearator open heater, and economizer. Closed feedwater heaters are shell-and-tube-type recuperators in which the feedwater temperature increases as the extraction steam condenses on the outside of the tubes carrying the feedwater. Since the two streams do not mix, they can be at different pressures. The dearator open heater is a direct contact-type heat exchanger performing a twofold function. First, extraction steam and feedwater streams, at different temperatures and pressures, mix to form a feedwater stream at an intermediate temperature. Second, operation at higher than atmospheric pressure is also used to vent oxygen and other dissolved gasses out from the cycle to maintain the working fluid purity to minimize equipment corrosion. The economizer is a separate heat exchanger where combustion flue gasses, instead of steam, are used to increase feedwater temperature. With respect to Figure 2.3, feedwater regenerative heating takes place along what is known as the feedwater path, comprising points 1 through 7. Isentropic adiabatic compressions of the

Figure 2.4 Temperature-entropy diagram of Rankine cycle.
working fluid are carried out by the condenser pumps (points and states 1-2), and by the feedwater pump (points and states 4-5). Isobaric increases in temperature and entropy take place at the low-pressure feedwater heaters (points and states 2-3), dearator (points and states 3-4), high-pressure feedwater heaters (points and states 5-6), and economizer (points and states 6-7).

Superheating takes the saturated steam coming out of the boiler drum (point 8) to a superheated vapor condition before entering the high pressure section of the steam turbine (point 9). Superheating usually takes place at high pressure in a set of separate heat exchangers, called primary and secondary superheaters, using combustion flue gasses. Steam temperature is increased at constant pressure from state 8 to state 9. After the isentropic expansion through the high-pressure turbine (points 9 to 10), the steam is taken again to a superheated vapor condition before entering the low-pressure turbine section (point 11). This second increase of temperature is carried out by the so called reheater using combustion flue gasses, and takes place at constant pressure with an intermediate value between those at the drum output and the condenser. Steam temperature increases from state 10 to 11. Finally, the steam undergoes another isentropic expansion through the low-pressure turbine section, developing mechanical work and losing temperature and pressure until the condenser values (point and state 12).

After all previous additions to the simple cycle, the thermal efficiency of the cycle may be calculated in terms of the heat energy content of the working fluid at the different points in Figure 2.3 (states in Figure 2.4). Defining convenient control volumes around the plant components, making the same assumptions as before, and applying the conservation laws of mass and energy, the overall plant efficiency is given by:

$$\eta = \frac{\dot{W}_{\text{hp}} / \dot{m} + \dot{W}_{\text{lp}} / \dot{m} - \dot{W}_{\text{pf}} / \dot{m} - \dot{W}_{\text{pc}} / \dot{m}}{Q_{\text{sh}} / \dot{m} + Q_{\text{pf}} / \dot{m} + Q_{\text{ec}} / \dot{m} + Q_{\text{sh}} / \dot{m}}$$  \hspace{1cm} (2.10)

where the work developed per unit mass at the high-pressure turbine section, low-pressure turbine section, feedwater pump, and condensate pump, respectively, may be approximated (neglecting steam extractions) by:

$$\dot{W}_{\text{hp}} / \dot{m} \equiv (h_1 - h_{10})$$  \hspace{1cm} (2.11a)
and the heat added at the superheaters, furnace waterwall risers, economizer, and reheater is given, respectively, by

\[ \dot{W}_{\text{sh}} / \dot{m} \equiv (h_{f} - h_{s}) \]  
\[ \dot{W}_{\text{fr}} / \dot{m} \equiv (h_{g} - h_{f}) \]  
\[ \dot{W}_{\text{ec}} / \dot{m} \equiv (h_{s} - h_{g}) \]  
\[ \dot{W}_{\text{rh}} / \dot{m} \equiv (h_{i} - h_{s}) \]  

(2.12a)  
(2.12b)  
(2.12c)  
(2.12d)

Since all previous expressions are obtained assuming ideal conditions, the significance of the ideal thermodynamic Rankine cycles in Figures 2.2 and 2.4 and the expressions (2.6) and (2.10) is that they establish the highest theoretical limit on plant performance, as imposed by plant design. Then, it seems logical that some insight for process operation betterment can be gained through comparison of the actual power unit performance against the highest theoretical limit. However, as will be shown in the next section, there are some unsolvable difficulties in pointing out where the improvements should be made using only the kind of efficiency information just provided.

### 2.1.3 Exergy analysis of a fossil fuel power unit

For a very long time, electric utilities have used the heat-rate method [Salisbury 1958, Salisbury 1959, Salisbury 1961] to evaluate the actual performance of power units. This method is deep-rooted on the application of the principles of conservation of mass and energy, as given in the previous section, and makes use of specification data and correction curves provided by the plant equipment manufacturers, as well as previous plant performance data, to assess the current plant thermal efficiency. However, this
method is not entirely satisfactory, since it is known that performance analysis based on the first law of thermodynamics provides only the quantities of energy transferred to and from the plant, and nothing can be said about the relative significance of the energy losses at the different plant components. As an example, first-law analysis indicates that the condenser greatly affects the power unit efficiency because of the large amount of heat transferred to the cooling water, without providing any clue on the real usefulness of this relatively low-temperature fluid, and could mislead any attempt to improve the process operation in this direction. Also, in general, energy balances do not provide information about internal losses such as in a throttling valve, or heat exchanger, which may give the wrong impression that these processes are free of losses.

Exergy analysis [Kotas 1995, Moran 1989] can provide the means for a more meaningful assessment of power unit performance [Gaggioli, et al., 1975]. Based on the concept of exergy, as the measure of quality or potential to do useful work of a given form of energy, and the application of the second law of thermodynamics, in the form of the increasing entropy principle, the results of an exergy analysis, as a breakdown of the loss of exergy, or irreversibility, among the components of a plant, can be used in many different ways, i.e., to optimize the operation of the power unit, or to evaluate each component condition for maintenance planning purposes. In spite of all its conveniences and versatility, exergy analysis has not been yet widely adopted as a routine procedure by electric utilities for performance auditing. Of particular interest for this work is the application of exergy analysis to establish an optimal operation strategy for wide range operation of a power unit, and to find out the way to achieve it through process control, which otherwise cannot be established quantitatively with first-law analysis only.

Irreversibility is the thermodynamic effect taking place in an irreversible process. An irreversible process in a system is one in which the system and its surroundings cannot be exactly restored to their initial state once the process has occurred. By contrast, in a reversible process the initial state can be recovered. Equivalent definitions state that a process in which the initial state is changed by applied work, and which can produce the same amount of work by returning to its original state, is a reversible process, otherwise
it is an irreversible process. Experimental observation demonstrates that all real processes are irreversible. In the process industries there are three main general classes of irreversible operations (irreversibilities) of interest: (1) mixing of fluids, (2) transfer of heat, and (3) flow of fluids. Reversible processes are theoretical idealizations to ease mathematical analysis. Quantification of the irreversibility of a spontaneous process in an isolated system may be done in terms of the entropy increase that always takes place, as stated by the second law of thermodynamics, in the direction from a more organized form of energy (lower entropy) to a form with greater randomness (higher entropy).

The rate of irreversibility change in a control volume, $I_{cv}$, per unit of mass flowing through the control volume, $m$, in an adiabatic process taking place at zero state temperature, $T_0$, with entropy increasing from $s_1$ at the single input to $s_2$ at the single output, is given by:

$$\frac{I_{cv}}{m} = T_0 (s_2 - s_1)$$

(2.13)

Exergy is the maximum theoretical work obtainable from the combined system formed by a closed system and its environment, as the closed system passes from a given initial state to the dead state while interacting with the environment only. Alternatively, exergy can be regarded as the magnitude of the minimum theoretical work input required to bring the closed system from the dead state to the original initial state. Once the environment is specified, exergy can be quantified in terms of the closed system property values. The exergy of a closed system can be calculated using energy and entropy balances. Although an extensive property, exergy can be expressed as an specific property on a unit mass. The specific exergy of a unit mass flow across a control volume is given by:

$$\varepsilon = (h - h_0) - T_0 (s_2 - s_1) + \frac{V^2}{2} + g z$$

(2.14)

where $\varepsilon$ stands for specific exergy in units of specific energy, $h$ and $s$ represent the specific enthalpy and entropy, respectively, at the control volume inlet or outlet under consideration; $h_0$ and $s_0$ are the same properties at the zero state. $V$ is the volume of the
closed system, $g$ is the acceleration of gravity, and $z$ is the elevation over the zero state. A steady-state exergy balance relationship may be obtained for a control volume with a single input and a single output:

$$0 = \sum_j \left( 1 - \frac{T_0}{T_j} \right) \dot{Q}_j - \dot{W}_{cv} + \dot{m}(e_i - e_o) - \dot{I}_{cv}$$

where the first three terms in the right correspond to rates of exergy transfer, and the last term is the rate of exergy destruction or irreversibility. The variable $\dot{Q}_j$ is the rate of heat transfer at a location on the boundary where temperature is $T_j$, and the term $\dot{W}_{cv}$ represents the rate of energy transfer by work other than flow work.

Exergy efficiency uses the concept of exergy in assessing the effectiveness in energy resource utilization. For a closed system receiving a heat transfer rate $\dot{Q}_s$ at a source temperature $T_s$, delivering a heat transfer rate $\dot{Q}_u$ at a user temperature $T_u$, and losing a heat transfer rate $\dot{Q}_l$ at a temperature $T_l$, the exergy efficiency is given by:

$$\eta_e = \frac{(1 - T_0 / T_s) \dot{Q}_s}{\eta (1 - T_0 / T_u)}$$

where $\eta_e$ is the exergy efficiency, and $\eta$ is the first law efficiency. Relationships (2.13) through (2.16) can be used to assess the plant performance, along the working fluid power cycle, by applying them successively to all control volumes under consideration.

Energy and exergy analysis have been carried out in actual power plants, with results leading to different conclusions [Sciubba and Su 1985]. From the first-law point of view, the working fluid thermal cycle efficiency is typically a bit above 30%, with a little bit below 70% of the energy supplied to the cycle being carried out by the condenser cooling water. Also, losses due to heat transfer with the surroundings and the exiting stack gases typically account for a great percentage of the supplied raw fuel energy. By contrast, second-law analysis shows that since the condenser cooling water temperature is raised only a few degrees over that of the surroundings, the exergy loss is considerably minor, accounting for only 1% of the total exergy provided by the fuel to the plant. The same can be said about the stack losses. Instead, exergy analysis calls the attention to a great exergy destruction (60%) in the boiler by irreversibilities in the furnace during fuel
combustion and heat transfer to vaporize water. Also, exergy destruction is significant at the turbine (5%) and at the condenser (3%). The results of exergy analysis highlight the role of water regeneration (feedwater heating) and steam reheat to increase the plant efficiency [Habib and Zubair 1992], in addition to increase the main steam temperature and pressure, as suggested by first-law analysis.

The above concepts are primarily concerned with the analysis of power plant performance for power plant design at steady-state conditions. Even if that constitutes an advance over current design practices, still fewer attempts have been made to apply the results of exergy analysis to the design of operation procedures and control systems for wide-range operation. The relevance of this subject is that due to the large costs of plant operation, maintenance, and fuel, even a very small (~1%) performance improvement will produce very significant savings. In this regard, [Habib, et al. 1995] showed how the exergy efficiency decreases during operation at lower than base loads. Exergy losses are traced down to the steam turbines, with the high pressure turbine accounting for the most part. It is pointed out that the sharp decay in efficiency of the high-pressure turbine is partly due to steam throttling at low loads. High pressure irreversibility losses increase by 2% as the load decreases from 100% (base load) to 25% load. In [Yasni and Carrington 1987] an experimental exergy audit was carried out in a 250 MW power unit working at 150 MW to compare plant performance under constant pressure and sliding pressure control modes. The superior efficiency of sliding pressure over constant pressure operation is shown by the Grassman diagrams of the exergy flows in Figures 2.5 and 2.6, respectively. One of the most significant facts is that improved performance, from 36.64% to 37.5%, was obtained by reducing the main steam throttling losses from 1.81% to 0.17% through control action over the main steam control valve.

Earlier, [Buchwald 1975] showed that the energy dissipated in a process can be reduced by lowering the pressure drop across the control valves, so that the energy required to operate many processes can be significantly reduced. The explanation can be found in [Moran and Saphirou 1996], where it is shown, through a simple example, how an adiabatic throttling process in a control valve preserves energy, but destroys exergy.
Figure 2.5 Exergy audit at constant pressure.
Reprinted with permission [Yasni and Carrington 1987]

Figure 2.6 Exergy audit at sliding pressure.
Reprinted with permission [Yasni and Carrington 1987]
From the above discussion it is clear that any wide-range process optimization attempt, through control actions, in a power plant should consider minimization of exergy losses in the control valves that regulate the flow of the working fluid as a very basic requirement. This in turn, can be utilized advantageously to reduce raw fuel consumption complementarily to process optimization through plant design.

2.1.4 Overall operation of power units

A power system is intended to supply the electric power demanded by the consumers in a reliable form with high quality characteristics. Total system load is not under direct control and follows daily, weekly, and seasonally cyclic patterns. Connection and disconnection of individual loads cause random fluctuations about these patterns. Also, since there are no practical means to store large quantities of electric energy, it should be produced as needed by the consumers. As a consequence, the power system never really operates in steady state; it is always trying to match power generation with the load in what is known as the load-frequency problem. Thus, FFPUs participating in load-frequency control are always subject to changing load demands and load disturbances as part of their normal operation regime [Dunlop and Ewart 1975].

At a glance, the qualitative operation of a FFPU is as follows. First, the power needed by the load, at a given frequency, determines the counteracting electromagnetic torque that is to be equalized by the actuating mechanical torque produced by the steam turbine as power at a given speed. So, frequency is used to determine the generation-load balance (mechanical-electrical torque balance). Frequency above the nominal value (e.g., 60 Hz) indicates that generation is larger than the load, frequency below the nominal value indicates that generation is below the load, and frequency at the nominal value indicates that the generation matches the load. Second, the prime mover torque is a function of the steam flow energy into the turbine, which in turn is a function of the steam pressure and temperature. Hence, the steam pressure before the throttle valves is
used to determine the boiler-turbine energy balance. If the throttle pressure increases, the boiler is producing more steam than the turbine requires. If the throttle pressure decreases then the boiler is producing less steam than required. When the throttle pressure is constant the boiler is matching the steam needed by the turbine. Third, the rate at which steam is generated is determined by the rate at which fuel is burned. Adjusting the firing rate requires adjusting both the fuel flow and the air flow to maintain safe and complete combustion. The rate at which steam is consumed is determined by the turbine valve position and the throttle pressure. At all moments feedwater should be supplied in adequate quantities to sustain the steam flow.

In summary, operators in a FFPU must satisfy two global operating requirements: generate the MW needed to satisfy the load demand, and maintain the energy balance within the unit. They satisfy these requirements supervising, through the BTG (Boiler-Turbine-Generator) board instrumentation, the adequate regulation of generated power, main steam pressure, and drum water level. To do so, they adjust the main steam flow, fuel and air flows, and feedwater flow. There are many other variables that need to be regulated in accordance to the generated power and operation practices, nevertheless the variables just mentioned account for the dominant behavior of a FFPU.

### 2.1.5 Qualitative dynamics of a power unit

From the power system perspective the overall input-output behavior of a FFPU has noteworthy relevance. On one hand, long-term frequency stability analysis, which assumes that all electromechanical oscillations have died out and that the system is operating at constant frequency, perhaps different from the nominal value, could be in a time frame of several to tens of minutes. On the other hand, the main boiler dynamics are relatively slow: steam pressure and temperature oscillations, and the effect of fuel flow variations on the generated power are in the order of minutes. Therefore, the dynamics of FFPUs are considered a major factor in frequency stability analysis [Kundur 1997].
Accordingly, any FFPU participating in load-frequency control duties should be equipped with control systems that take into account the long-term overall input-output dynamic behavior of the unit.

As explained before, the electric power in a drum-type FFPU is the resultant of a series of energy conversion processes within the unit. All those energy conversion processes are rather complex and show very complex relationships among them. However, the essential overall dynamics may be described in terms of the major inputs (fuel flow, air flow, steam flow into the turbine, feedwater flow, and spray flows into the superheater and reheater) and outputs (electric power, steam throttle pressure, drum water level, superheater outlet temperature, and reheater outlet temperature) [Maffezoni 1996]. Electric power and steam pressure are tightly coupled and are affected heavily by the fuel/air flow and the steam flow. Feedwater flow slightly affects power and pressure, but greatly affects the drum level, which in turn is considerably affected by the fuel and steam flows. Similarly, the spray flows have a minor effect on power and pressure, but greatly affect the heaters outlet temperatures, which are heavily influenced by the fuel flow. In summary, fuel and steam flow may be used to drive the unit to the desired values of power and pressure, this will disturb the drum water level and heaters outlet temperatures, which may then be manipulated with the feedwater and spray flows.

The interaction among fuel, steam, and feedwater flows as inputs, and power, pressure and water level as outputs, obligates to primarily take these variables into account to achieve wide-range operation. Spray flows and temperatures can be used for further improvement. Consequently, this research project concentrates in the former situation. Furthermore, the open-loop behavior determines the input-output pairing to form the feedback control loops. Figure 2.7 shows the response to an opening-step in the steam valve with the fuel and feedwater valves kept constant. Power increases and decays back close to its original value, while pressure decreases to a new value and the level keeps decreasing. Figure 2.8 shows the response to an opening-step in the fuel valve, with the steam and feedwater valves at a fixed position. Both throttle pressure and power increase to a new fixed higher value, while the level keeps decreasing again.
Figure 2.7 Open-loop response to step in steam valve position.

Figure 2.8 Open-loop response to step in fuel valve position.
From these tests it can be seen that for short-term purposes, a fast response to load variations may be attained using the throttle valve to control power and the fuel valve to regulate the steam pressure. Conversely, for long-term purposes, fuel flow should be used to control power, and the throttle valve to control the steam pressure. In both cases the level has to be regulated to balance plant operation.

### 2.2 Mathematical Models of Power Units

In accordance to the scope of this research work, stated in Section 1.4, the main concern is the design of an overall control system for a fossil fuel drum-type power unit. Only the fundamental control strategy for optimal long-term wide-range normal dynamic operation is of interest, since it constitutes the kernel around which all other components of the control system are to be designed. In this regard, the use of a suitable process model to support the development of the control system is a critical issue for success. Nowadays, there is available a substantial number of mathematical models for fossil fuel drum-type power units in the technical literature, covering an ample range of possible applications.

Focus on operation optimality in the long term sets the requirement on a model with capacity to simulate long term dynamics in the order of several minutes to hours. In this time frame, the hosting power system may operate under mismatches between power generation and power consumption, possibly at different-from-nominal but nearly constant frequency. Consequently, it is neither meaningful nor necessary for the model to take into account the short-term and fast dynamics associated with all transient electromagnetic phenomena at the electric generator and the excitation system, as is usually required for voltage stability studies [Kundur 1994]. Instead, major emphasis must be made on the boiler dynamics, which are slow and have large duration. Perhaps the relatively fast inertial effects of the turbine-generator set should also be considered.
Practically, nonlinear models are mandatory in order to be able to satisfy power unit wide-range operation requirements, since linear small perturbation models are normally only valid in a small and usually unknown neighborhood around a fixed point of operation, without any regard to the technique used to find them. Nonlinear models obtained from first-principles may not only be used for system simulation along the desired wide-range operation trajectories, but also to investigate and predict the process behavior around them after major events on the system, as required for the design of protection systems.

Interest on overall operation and dynamics inherently carries out a trade off on model complexity. Large models with very detailed models for plant components are adequate for training simulators, where close resemblance to actual power plants is a must. Also, with the adequate interfaces, and real-time execution, these models may be used to evaluate and pre-tune mature control systems in an advanced stage of a control system development project, just before application in the actual plant. However, the design of the overall control strategy asks for the essentials to be considered. In this regard, it is of prime importance to find out the meaningful process inputs, outputs and internal variables, and to capture and reproduce the nonlinear and interaction effects in close resemblance to the physical process, in such a way that clear insight on the significant overall behavior of the process can be gained for control system design.

2.3 FFPU Non-Linear Model

The essential overall dynamics of a FFPU have been remarkably captured for a 160 MW oil fired drum-type boiler-turbine-generator unit in a third order multi-input-multi-output nonlinear model by [Bell and Åström 1987b]. The model represents the boiler unit P16 and the turbine unit G16 at Oresundsverket in Malmo, Sweden, and has been an active research topic for the last 3 decades. It can be considered the best simple model currently available for simulation of the overall dynamics of a FFPU.
The first version of this model appeared in [Åström and Eklund 1972, Åström and
Eklund 1975], which modeled the electric power output and drum pressure by a 2\textsuperscript{nd} order
nonlinear model. Later, in [Bell and Åström 1979] additions were made to predict the
 drum water level, increasing the model to 7\textsuperscript{th} order. Also, actuator dynamics for the fuel,
 feedwater and throttle control valves were included. In [Bell and Åström 1987a] two
 versions of the model were proposed, the main issue being the integration of evaporation
 rate concept by [Morton and Price 1977] which simplified the drum level prediction and
 the model reduced back to the 3\textsuperscript{rd} order. In [Bell and Åström 1987b] the results of a
 comparative study among several low order model were presented, where the 3\textsuperscript{rd} order
 model performed satisfactorily. More recently, in [Åström and Bell 1993] the modeling
 of the drum water level was supported in physical first principles without increasing the
 model order. Finally, in [Bell and Åström 1996] the drum boiler part of the model was
 revisited and increased to 4\textsuperscript{th} order to account more precisely for the shrink and swell
 effect in the drum water level. In what follows, a summary of the 3\textsuperscript{rd} order model used in
 this work is presented since the original is dispersed in some of the references previously
 provided.

\subsection*{2.3.1 Mathematical model}

The FFPU is represented by a third order nonlinear model with three state
equations, three inputs, and three outputs. The inputs are the positions of the valve
actuators that control the mass flow rates of fuel ($u_1$ in pu), steam to the turbine ($u_2$ in pu),
and feedwater to the drum ($u_3$ in pu). The three outputs are the electrical power ($E$ in
MW), drum steam pressure ($P$ in kg/cm\textsuperscript{2}), and drum water level deviation ($L$ in m). The
three state variables are the electric power, drum steam pressure, and fluid (steam-water)
density ($\rho_f$).

Overall modeling of the power unit starts by considering the boiler-turbine system
as an energy reservoir. Energy, $P_i$, enters the boiler-turbine by means of the fuel and the
feedwater being fed. Energy, $P_o$, leaves the boiler-turbine in the form of mechanical energy. Energy, $H$, is stored in the iron, liquid water, and steam masses. An overall energy balance yields the rate of stored energy as:

$$\frac{dH}{dt} = P_i - P_o$$  \hspace{1cm} (2.17)$$

Simple and reasonable good expressions for $H$, $P_i$, and $P_o$ were found based on experimental data. For the energy storage, the major considerations were: 1) the distribution of energy stored in iron, water and steam masses do not change during transients, so any energy dependent variable could be used as a measure of stored energy, and 2) the temperatures of boiler pipes usually are not measured, turbine inlet temperatures are kept constant over wide operation ranges, and drum pressure change significantly, so drum pressure is used as the measure of energy storage. The energy storage is approximated as a linear function of the drum pressure, $P$, with constants $a$ and $b$ to be determined, as:

$$H = H(P) = aP + b$$  \hspace{1cm} (2.18)$$

To find out an expression for the input power, $P_i$, it was assumed that the efficiency of the boiler is constant, thus implying that the air flow during fast changes of the fuel flow always can be manipulated to meet the demand. Then, the input power depends only on the fuel flow, which is proportional to the fuel control valve position, $u_1$, and the feedwater flow, which is proportional to the feedwater control valve position, $u_3$. Moreover, power from feedwater is proportional to the product of the feedwater flow and the enthalpy difference of water at feedwater and saturation states in the drum. Assuming that this enthalpy difference is constant yields:

$$P_i = a_1u_1 - a_2u_3$$  \hspace{1cm} (2.19)$$

where $a_1$ and $a_2$ are constants to be determined.

To find out the output power, $P_o$, the energy effects due to pressure drop across the steam heaters and the turbine are neglected, thus being proportional to the product of the mass steam flow, $w_s$, and the enthalpy drop across the turbine, $\Delta h$:

$$P_o = b_1w_s\Delta h - b_2$$  \hspace{1cm} (2.20)$$
where \( b_1 \) and \( b_2 \) are constants to be determined. In turn, the steam mass flow is proportional to the product of the throttle valve position, \( u_2 \), and the drum pressure, \( P \), and the enthalpy drop is proportional to the drum pressure to the 1/8 power. Then \( P_o \) is expressed as:

\[
P_o = \alpha_4 (u_2 P^{9/8} - \alpha_5)
\]

(2.21)

where \( \alpha_4 = 0.6 \) and \( \alpha_5 = 0 \) are constants determined from experimental data.

Taking the derivative of (2.18), equating to (2.17), substituting (2.19) and (2.21), and solving for the rate of drum pressure, with the constant fitted to the experimental data, provides the state equation:

\[
\frac{dP}{dt} = 0.9u_1 - 0.0018u_2 P^{9/8} - 0.15u_3
\]

(2.22)

Another state equation is obtained from an energy balance for the turbine-generator system. In steady state, the generator energy output, \( E \), is proportional to the power produced by the turbine as given by (2.21). However, in this case the steam flow is compensated for the energy passing to the condenser and the feedwater heaters. With the constant values chosen to fit experimental data, the energy balance yields:

\[
E = (0.73u_2 - 0.16) P^{9/8}
\]

(2.23)

Further inclusion of the turbine-generator inertia dynamics yields the desired state equation for the electric power:

\[
\frac{dE}{dt} = \left( [0.73u_2 - 0.16] P^{9/8} - E \right) \frac{1}{10}
\]

(2.24)

The third state equation is motivated by the calculation of the drum water level deviation, \( L \), as a linear approximation of a nonlinear function of the volume of water, \( V_w \), in the drum below the base reference value:

\[
L = c_1 V_w + c_2
\]

(2.25)

where \( c_1 = 50 \) and \( c_2 = 65.5 \) are found from the drum dimensions, and \( V_w \) may be expressed as the sum of the total amount of water in the drum, \( V_{w_o} \), and the amount of evaporating steam, \( V_e \). The rate of total water in the drum is determined by the feedwater mass flow rate, \( w_{fw} \), the steam mass flow rate, \( w_s \), and the specific volume of water at the given operating conditions, \( v_w \):
Since \( V_{wa} \) may be expressed as \( V_{wa} = V_{t} \rho_{f} \), where \( V_{t} \) stands for the drum volume, and \( \rho_{f} \) is the fluid (water-steam) density, (2.26) yields the third state equation when the feedwater flow, \( w_{fw} \), is given in terms of the feedwater valve position, \( u_{3} \), and the steam flow, \( w_{s} \), in terms of the steam valve position, \( u_{2} \), and the drum pressure, \( P \):

\[
\frac{d \rho_{f}}{dt} = \frac{(141u_{3} - (1.1u_{2} - 0.19)P)}{V_{t}}
\]  (2.27)

Additional algebraic equations are required to calculate the drum water level from (2.25). All model equations are summarized below:

\[
\frac{dE}{dt} = \left( (0.73u_{2} - 0.16)P^{9/8} - E \right)/10
\]  (2.28)

\[
\frac{dP}{dt} = 0.9u_{1} - 0.0018u_{2}P^{9/8} - 0.15u_{3}
\]  (2.29)

\[
\frac{d \rho_{f}}{dt} = (141u_{3} - (1.1u_{2} - 0.19)P)/V_{t}
\]  (2.30)

\[
w_{f} = k_{f}u_{1}
\]  (2.31)

\[
w_{s} = (1.1u_{2} - 0.19)P
\]  (2.32)

\[
w_{fw} = k_{fw}u_{3}
\]  (2.33)

\[
e_{f} = k_{f1}w_{f} + k_{f2}
\]  (2.34)

\[
w_{e} = (k_{e}e_{f} - rw_{fw} + Kw_{s}) / (1 + K)
\]  (2.35)

\[
\rho_{s} = k_{s1}P + k_{s2}
\]  (2.36)

\[
\alpha_{s} = (1 / \rho_{f} - v_{w}) / (1 / \rho_{s} - v_{w})
\]  (2.37)

\[
L = 50(\rho_{f}v_{w} + 60\alpha_{s} + T_{s}V_{f}w_{e} / a - 65.5)
\]  (2.38)
Table 2.1 summarizes all variables used in the model, Table 2.2 summarizes all constant parameters used in the model, and Table 2.3 provides the base load operating conditions.

**TABLE 2.1 FFPU MODEL VARIABLES**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u_1$</td>
<td>Fuel valve actuator position</td>
<td>pu</td>
</tr>
<tr>
<td>$u_2$</td>
<td>Throttle valve actuator position</td>
<td>pu</td>
</tr>
<tr>
<td>$u_3$</td>
<td>Feedwater valve actuator position</td>
<td>pu</td>
</tr>
<tr>
<td>$E$</td>
<td>Electric power output</td>
<td>MW</td>
</tr>
<tr>
<td>$P$</td>
<td>Drum steam pressure</td>
<td>Kg/cm$^2$</td>
</tr>
<tr>
<td>$L$</td>
<td>Drum water level</td>
<td>m</td>
</tr>
<tr>
<td>$\rho_f$</td>
<td>Drum/raiser fluid (steam, water) density</td>
<td>Kg/m$^3$</td>
</tr>
<tr>
<td>$w_f$</td>
<td>Fuel mass flow rate</td>
<td>Kg/s</td>
</tr>
<tr>
<td>$w_s$</td>
<td>Steam mass flow rate</td>
<td>Kg/s</td>
</tr>
<tr>
<td>$w_w$</td>
<td>Feedwater mass flow rate</td>
<td>Kg/s</td>
</tr>
<tr>
<td>$e_f$</td>
<td>Fuel heat rate</td>
<td>Kw</td>
</tr>
<tr>
<td>$\rho_s$</td>
<td>Steam density</td>
<td>Kg/m$^3$</td>
</tr>
<tr>
<td>$\alpha_s$</td>
<td>Steam quality</td>
<td>---</td>
</tr>
<tr>
<td>$w_e$</td>
<td>Evaporation mass flow rate</td>
<td>Kg/s</td>
</tr>
</tbody>
</table>

**TABLE 2.2 FFPU MODEL CONSTANT PARAMETERS**

<table>
<thead>
<tr>
<th>Param</th>
<th>Value</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_t$</td>
<td>85.0</td>
<td>m$^3$</td>
<td>Total volume of drum, downcomers and risers</td>
</tr>
<tr>
<td>$v_w$</td>
<td>0.001538</td>
<td>m$^3$/Kg</td>
<td>Specific volume of water at 320 °C</td>
</tr>
<tr>
<td>$k_{f0}$</td>
<td>12.6</td>
<td>Kg/s</td>
<td>Proportional constant for fuel flow 50x0.252</td>
</tr>
<tr>
<td>$k_{f1}$</td>
<td>20200</td>
<td>Kw</td>
<td>Proportional constant for fuel heat power (energy)</td>
</tr>
<tr>
<td>$k_{f2}$</td>
<td>-11700</td>
<td>Kw</td>
<td>Bias constant for fuel heat power</td>
</tr>
</tbody>
</table>
\[ k_{fw} \quad 126.0 \quad \text{Kg/s} \quad \text{Proportional constant for feedwater flow} \]

\[ K \quad 3.46608 \quad --- \quad \text{Additional mass of steam evaporated per unit mass lost from the system, } K = k_a k_b k_c \]

\[ k_b \quad 0.0008 \quad \text{Kg/KJ} \quad \text{Reciprocal of latent heat of vaporization (1/h_{fg})} \]

\[ r \quad 0.08912 \quad --- \quad \text{Loss of fuel energy used for evaporation per unit mass of feedwater entering (h_r-h_w)/h_{fg}} \]

\[ k_{s1} \quad 0.8 \quad \text{cm}^2/\text{m}^3 \quad \text{Proportional constant from fitting data to steam tables} \]

\[ k_{s2} \quad -25.6 \quad \text{Kg/m}^3 \quad \text{Bias constant from fitting data to steam tables} \]

\[ T_s \quad 2000 \quad \text{s m}^2 \quad \text{Fall of boiler water mass per unit increase in evaporation rate} \]

\[ V_f \quad 0.0015 \quad \text{m}^3/\text{Kg} \quad \text{Specific volume of saturated water (constant) from steam tables at 322 °C} \]

\[ a \quad 27.0 \quad \text{m}^2 \quad \text{Drum water surface area at normal operating level} \]

---

### TABLE 2.3 FFPU RATED OPERATING CONDITIONS

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electric power</td>
<td>160</td>
<td>MW</td>
</tr>
<tr>
<td>Steam flow</td>
<td>140</td>
<td>Kg/sec</td>
</tr>
<tr>
<td>Drum steam pressure</td>
<td>140</td>
<td>Kg/cm(^2)</td>
</tr>
<tr>
<td>Superheated steam temperature</td>
<td>535</td>
<td>°C</td>
</tr>
<tr>
<td>Feedwater temperature</td>
<td>300</td>
<td>°C</td>
</tr>
<tr>
<td>Volume of drum</td>
<td>40</td>
<td>m(^3)</td>
</tr>
<tr>
<td>Volume of downcomers</td>
<td>11</td>
<td>m(^3)</td>
</tr>
<tr>
<td>Volume of raisers</td>
<td>38</td>
<td>m(^3)</td>
</tr>
<tr>
<td>Mass of water in system</td>
<td>40,000</td>
<td>Kg</td>
</tr>
<tr>
<td>Mass of steam in system</td>
<td>2,000</td>
<td>Kg</td>
</tr>
</tbody>
</table>
2.3.2 Actuators

Control valves are modeled as shown in Figure 2.9. Their characteristics are given in table 2.4.

![Figure 2.9 Valve actuator block diagram.](image)

Notes:
- Implementation of actuator dynamics was made after the model given in [Bell and Åström 1979] with unit gain and programmable time constant.
- Model in appendix E of [Bell and Åström 1987b] was not used because the integration to find the bounded output may stop before reaching the input value if the time constant is small.
- Time constant values given in table 5 were determined from simulation experiments.

<table>
<thead>
<tr>
<th>Valve</th>
<th>Position (pu)</th>
<th>Rate (pu/sec)</th>
<th>Time constant (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fuel</td>
<td>$0 \leq u_1 \leq 1$</td>
<td>$-0.007 \leq du_1/dt \leq 0.007$</td>
<td>0.1</td>
</tr>
<tr>
<td>steam</td>
<td>$0 \leq u_2 \leq 1$</td>
<td>$-2 \leq du_2/dt \leq 0.02$</td>
<td>0.1</td>
</tr>
<tr>
<td>feedwater</td>
<td>$0 \leq u_3 \leq 1$</td>
<td>$-0.05 \leq du_3/dt \leq 0.05$</td>
<td>0.1</td>
</tr>
</tbody>
</table>
2.4 FFPU Static Models

The static model of the FFPU is used to calculate equilibrium points, determine operating windows, and to generate set-points. The static equations can be solved in different ways. Some of them are:

- **Direct static model.** Static equations solved to calculate the process output signals, with input signals as independent variables.
- **Inverse static model.** Static equations solved to calculate the process input signals, with the output signals as independent variables.
- **Mixed static model.** Static equations solved to calculate a mixture of process inputs and outputs signals, with the remaining input and output signals as independent variables.

In steady-state the state variables are constant, thus,

\[
\frac{dE}{dt} = \frac{dP}{dt} = \frac{dp}{dt} = 0
\]

(2.39a)

\[
L = 0
\]

(2.39b)

Substitution into the dynamic model equations yields:

\[
0 = 0.9u_1 - 0.0018u_2P^{9/8} - 0.15u_3
\]

(2.40a)

\[
0 = (0.73u_2 - 0.16)P^{9/8} - E
\]

(2.40b)

\[
0 = 14u_3 - (1.1u_2 - 0.19)P
\]

(2.40c)

The algebraic model equations become:

\[
0 = V_i \rho_f + 60\alpha_2 + T_j V_j w / \alpha - 65.5
\]

(2.41a)

\[
\alpha_2 = (1/ \rho_f - v_w) / (1/ \rho_i - v_w)
\]

(2.41b)

\[
\rho_s = k_{s1}P + k_{s2}
\]

(2.41c)

\[
w_e = (k_b (k_{f1}k_{f0}u_1 + k_{f2}) + K(1.1u_2 - 0.19)P - r k_{f0}u_3) / (1 + K)
\]

(2.41d)
Substitution of numerical values into the algebraic equations gives:

\[
0 = 0.13073 \rho_f + 60 \alpha_s + w_e / 9 - 65.5 \quad \text{(2.42a)}
\]

\[
\alpha_s = (1 / \rho_f - 0.00154) / (1 / \rho_s - 0.00154) \quad \text{(2.42b)}
\]

\[
\rho_s = 0.8P - 25.6 \quad \text{(2.42c)}
\]

\[
w_e = 45.59166u_1 + (0.8537u_2 - 0.14746)P - 2.51431u_3 - 2.0958 \quad \text{(2.42d)}
\]

### 2.4.1 Direct static model

The direct static model solves the static equations to determine the process outputs \((E, P, L)\) and state variables \((E, P, \rho_f)\) given the values of the input signals \((u_1, u_2, u_3)\). Since \(L = 0\) in steady state, it is only necessary to determine \(E, P, \) and \(\rho_f\).

From (2.40a) and (2.40b):

\[
E = \frac{0.73u_2 - 0.16}{0.0018u_2} (0.9u_1 - 0.15u_3) \quad \text{(2.43)}
\]

From (2.40c):

\[
P = \frac{141u_3}{1.1u_2 - 0.19} \quad \text{(2.44)}
\]

The working fluid (steam-water) density \(\rho\) is determined by solving the second order equation formed by substituting equation (2.41b) into (2.41a):

\[
0 = v_w V_f \rho_f^2 + \left( T_s w_e / a - 65.5 - \frac{v_w \rho_s}{1 - v_w \rho_s} \right) \rho_f + \frac{60 \rho_s}{1 - v_w \rho_s} \quad \text{(2.45)}
\]

where \(\rho_s\) and \(w_e\) are given by equations (2.41c) and (2.41d), respectively. Substitution of numerical values for constants in (2.45) yields:

\[
0 = 0.13073 \rho_f^2 + \left( \frac{w_e}{9} - 65.5 - \frac{0.00154 \rho_s}{1 - 0.00154 \rho_s} \right) \rho_f + \frac{60 \rho_s}{1 - 0.00154 \rho_s} \quad \text{(2.46)}
\]

where \(\rho_s\) and \(w_s\) are given by equations (2.42c) and (2.42d), respectively.
2.4.2 Inverse static model

The inverse static model solves the static equations to determine the process inputs \((u_1, u_2, u_3)\) and state variables \((E, P, \rho_f)\) given the values of the output signals \((E, P, L)\).

Equations (2.40a), (2.40b), and (2.40c) may be rewritten in matrix form as:

\[
\begin{bmatrix}
0.9 & -0.0018P^{9/8} & -0.15 \\
0 & 0.73P^{9/8} & 0 \\
0 & -1.1P & 141
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3
\end{bmatrix} =
\begin{bmatrix}
0 \\
0.16P^{9/8} + E \\
-0.19P
\end{bmatrix}
\]

(2.47)

with sequential solutions:

\[
u_2 = \frac{0.16P^{9/8} + E}{0.73P^{9/8}}
\]

(2.48a)

\[
u_3 = \frac{(1.1u_2 - 0.19)P}{141}
\]

(2.48b)

\[
u_1 = \frac{0.0018u_2P^{9/8} + 0.15u_3}{0.9}
\]

(2.48c)

The fluid (steam-water) density, \(\rho_f\), is calculated with (2.46), with \(\rho_s\) and \(w_e\) given by equations (2.42c) and (2.42d), respectively. Since (2.46) provides two solutions for the fluid density, only the value with a reasonable physical meaning is selected, as given in the next section.

According to equations (2.48a), (2.48b), and (2.48c), operation around rated power and pressure conditions (Table 2.3) may provide input values exceeding the high physical limits, specifically \(u_2 > 1\). Since the position of the valve actuators was collected manually some involuntary error was introduced, while all other measurements were made automatically. To deal with this situation in this research project, scaling factors are introduced in the command signals to the actuators to achieve rated operating conditions with control valve position values within the specified physical limits \([0,1]\). This course of action does not affect the basic formulation of the model, since it is equivalent to a valve replacement with resizing procedure in an actual power plant. The adjustment is
made on the base that current practice requires the control valves to also provide for control action at peak load (~110% base load) conditions. Thus, the scaling factor values for the input signals \( u_1, u_2, \) and \( u_3 \) are respectively chosen as:

\[
\begin{align*}
    k_{ssu_1} &= 0.7965973943 \quad (2.49a) \\
    k_{ssu_2} &= 1.1814399210 \quad (2.49b) \\
    k_{ssu_3} &= 1.1420265349 \quad (2.49c)
\end{align*}
\]

so that the power range spans from 10 MW (minimum load) through 180 MW (peak load), and the operating points in Table 2.5 are obtained successively:

**TABLE 2.5 OPERATING POINTS FOR INPUT SCALING**

<table>
<thead>
<tr>
<th>( E ) (MW)</th>
<th>( P ) (Kg/cm²)</th>
<th>( L ) (m)</th>
<th>( u_1 ) (p.u.)</th>
<th>( u_2 ) (p.u.)</th>
<th>( u_3 ) (p.u.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>150</td>
<td>0</td>
<td>1.0000</td>
<td>0.9293</td>
<td>0.9480</td>
</tr>
<tr>
<td>170</td>
<td>145</td>
<td>0</td>
<td>0.9484</td>
<td>0.9153</td>
<td>0.9000</td>
</tr>
<tr>
<td>160</td>
<td>140</td>
<td>0</td>
<td>0.8967</td>
<td>0.9000</td>
<td>0.8517</td>
</tr>
</tbody>
</table>

**2.4.3 Equilibrium points**

The inverse static model is used to determine several equilibrium points spanning the whole power unit operating range. Table 2.6 provides equilibrium points for a constant pressure operating policy, whereas Table 2.7 provides them for a typical sliding pressure operating policy. The relevance of these points will become obvious in the following chapters, where they will be used as reference points for comparison with other operating policies, and as initial conditions for process optimization procedures.
TABLE 2.6 CONSTANT PRESSURE EQUILIBRIUM POINTS

<table>
<thead>
<tr>
<th>$E$ (MW)</th>
<th>$P$ (kg/cm²)</th>
<th>$\rho_f$ (kg/m³)</th>
<th>$u_1$ (p.u.)</th>
<th>$u_2$ (p.u.)</th>
<th>$u_3$ (p.u.)</th>
<th>$L$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>140.0</td>
<td>457.0796</td>
<td>0.1999</td>
<td>0.2302</td>
<td>0.0949</td>
<td>0.0000</td>
</tr>
<tr>
<td>20.0</td>
<td>140.0</td>
<td>448.4947</td>
<td>0.2464</td>
<td>0.2748</td>
<td>0.1453</td>
<td>0.0000</td>
</tr>
<tr>
<td>40.0</td>
<td>140.0</td>
<td>431.0214</td>
<td>0.3393</td>
<td>0.3641</td>
<td>0.2462</td>
<td>0.0000</td>
</tr>
<tr>
<td>60.0</td>
<td>140.0</td>
<td>413.0712</td>
<td>0.4322</td>
<td>0.4534</td>
<td>0.3472</td>
<td>0.0000</td>
</tr>
<tr>
<td>80.0</td>
<td>140.0</td>
<td>394.5281</td>
<td>0.5251</td>
<td>0.5428</td>
<td>0.4481</td>
<td>0.0000</td>
</tr>
<tr>
<td>100.0</td>
<td>140.0</td>
<td>375.2257</td>
<td>0.6180</td>
<td>0.6321</td>
<td>0.5490</td>
<td>0.0000</td>
</tr>
<tr>
<td>120.0</td>
<td>140.0</td>
<td>354.9094</td>
<td>0.7109</td>
<td>0.7214</td>
<td>0.6499</td>
<td>0.0000</td>
</tr>
<tr>
<td>140.0</td>
<td>140.0</td>
<td>333.1566</td>
<td>0.8038</td>
<td>0.8107</td>
<td>0.7508</td>
<td>0.0000</td>
</tr>
<tr>
<td>160.0</td>
<td>140.0</td>
<td>309.1649</td>
<td>0.8967</td>
<td>0.9000</td>
<td>0.8517</td>
<td>0.0000</td>
</tr>
<tr>
<td>180.0</td>
<td>140.0</td>
<td>280.9982</td>
<td>0.9896</td>
<td>0.9893</td>
<td>0.9526</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

TABLE 2.7 SLIDING PRESSURE EQUILIBRIUM POINTS

<table>
<thead>
<tr>
<th>$E$ (MW)</th>
<th>$P$ (kg/cm²)</th>
<th>$\rho_f$ (kg/m³)</th>
<th>$u_1$ (p.u.)</th>
<th>$u_2$ (p.u.)</th>
<th>$u_3$ (p.u.)</th>
<th>$L$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>65.0</td>
<td>484.7999</td>
<td>0.1129</td>
<td>0.2914</td>
<td>0.0762</td>
<td>0.0000</td>
</tr>
<tr>
<td>20.0</td>
<td>70.0</td>
<td>475.7112</td>
<td>0.1659</td>
<td>0.3803</td>
<td>0.1323</td>
<td>0.0000</td>
</tr>
<tr>
<td>40.0</td>
<td>80.0</td>
<td>457.1753</td>
<td>0.2715</td>
<td>0.5208</td>
<td>0.2418</td>
<td>0.0000</td>
</tr>
<tr>
<td>60.0</td>
<td>90.0</td>
<td>437.9320</td>
<td>0.3765</td>
<td>0.6260</td>
<td>0.3485</td>
<td>0.0000</td>
</tr>
<tr>
<td>80.0</td>
<td>100.0</td>
<td>417.6470</td>
<td>0.4812</td>
<td>0.7071</td>
<td>0.4527</td>
<td>0.0000</td>
</tr>
<tr>
<td>100.0</td>
<td>110.0</td>
<td>395.8517</td>
<td>0.5854</td>
<td>0.7712</td>
<td>0.5549</td>
<td>0.0000</td>
</tr>
<tr>
<td>120.0</td>
<td>120.0</td>
<td>371.7977</td>
<td>0.6894</td>
<td>0.8229</td>
<td>0.6553</td>
<td>0.0000</td>
</tr>
<tr>
<td>140.0</td>
<td>130.0</td>
<td>344.0761</td>
<td>0.7932</td>
<td>0.8650</td>
<td>0.7542</td>
<td>0.0000</td>
</tr>
<tr>
<td>160.0</td>
<td>140.0</td>
<td>309.1649</td>
<td>0.8967</td>
<td>0.9000</td>
<td>0.8517</td>
<td>0.0000</td>
</tr>
<tr>
<td>180.0</td>
<td>150.0</td>
<td>242.5809</td>
<td>1.0000</td>
<td>0.9293</td>
<td>0.9480</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
2.4.4 Power-pressure operating window

The FFPU operating windows, or operating regions, are the sets of all feasible operating points for the FFPU. They can be determined through computer simulations using the nonlinear and the static models just described, taking into account equipment physical limitations (hard constraints) or operational limitations (soft constraints). An operating point is declared feasible when stable steady-state operation is achievable at that point, while all imposed constraints are satisfied. The operating windows can be used as the feasible regions of the decision variables during process optimization.

From the process optimization perspective, the power-pressure operating window is the most important one. Shown in Figure 2.10, it clearly indicates that any required power can be generated at any pressure value between the depicted upper and lower pressure limits. Tables 2.8 and 2.9 provide some numerical values of the operating points. Also shown in the same figure are the power-pressure relationships for the constant-pressure and the typical sliding-pressure operating policies that were provided in Tables 2.6 and 2.7, respectively.

![Figure 2.10 Power-pressure operating window.](image-url)
### TABLE 2.8 UPPER PRESSURE LIMIT EQUILIBRIUM POINTS

<table>
<thead>
<tr>
<th>$E$ (MW)</th>
<th>$P$ (kg/cm²)</th>
<th>$\rho_f$ (kg/m³)</th>
<th>$u_1$ (p.u.)</th>
<th>$u_2$ (p.u.)</th>
<th>$u_3$ (p.u.)</th>
<th>$L$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>237.4</td>
<td>318.1</td>
<td>0.3225</td>
<td>0.2102</td>
<td>0.1226</td>
<td>0.0000</td>
</tr>
<tr>
<td>20.0</td>
<td>231.8</td>
<td>312.0</td>
<td>0.3610</td>
<td>0.2362</td>
<td>0.1683</td>
<td>0.0000</td>
</tr>
<tr>
<td>40.0</td>
<td>220.9</td>
<td>300.8</td>
<td>0.4385</td>
<td>0.2965</td>
<td>0.2607</td>
<td>0.0000</td>
</tr>
<tr>
<td>60.0</td>
<td>210.2</td>
<td>290.1</td>
<td>0.5167</td>
<td>0.3551</td>
<td>0.3544</td>
<td>0.0000</td>
</tr>
<tr>
<td>80.0</td>
<td>199.7</td>
<td>278.9</td>
<td>0.5956</td>
<td>0.4251</td>
<td>0.4495</td>
<td>0.0000</td>
</tr>
<tr>
<td>100.0</td>
<td>189.5</td>
<td>268.1</td>
<td>0.6752</td>
<td>0.5032</td>
<td>0.5459</td>
<td>0.0000</td>
</tr>
<tr>
<td>120.0</td>
<td>179.5</td>
<td>257.8</td>
<td>0.7555</td>
<td>0.5907</td>
<td>0.6439</td>
<td>0.0000</td>
</tr>
<tr>
<td>140.0</td>
<td>169.7</td>
<td>247.1</td>
<td>0.8366</td>
<td>0.6890</td>
<td>0.7434</td>
<td>0.0000</td>
</tr>
<tr>
<td>160.0</td>
<td>160.2</td>
<td>236.8</td>
<td>0.9183</td>
<td>0.7996</td>
<td>0.7996</td>
<td>0.0000</td>
</tr>
<tr>
<td>180.0</td>
<td>150.9</td>
<td>226.3</td>
<td>1.0000</td>
<td>0.9245</td>
<td>0.9245</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

### TABLE 2.9 LOWER PRESSURE LIMIT EQUILIBRIUM POINTS

<table>
<thead>
<tr>
<th>$E$ (MW)</th>
<th>$P$ (kg/cm²)</th>
<th>$\rho_f$ (kg/m³)</th>
<th>$u_1$ (p.u.)</th>
<th>$u_2$ (p.u.)</th>
<th>$u_3$ (p.u.)</th>
<th>$L$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>32.0</td>
<td>493.0</td>
<td>0.0785</td>
<td>0.4205</td>
<td>0.0708</td>
<td>0.0000</td>
</tr>
<tr>
<td>20.0</td>
<td>32.0</td>
<td>485.2</td>
<td>0.1274</td>
<td>0.6554</td>
<td>0.1315</td>
<td>0.0000</td>
</tr>
<tr>
<td>40.0</td>
<td>36.3</td>
<td>468.8</td>
<td>0.2287</td>
<td>1.0000</td>
<td>0.2504</td>
<td>0.0000</td>
</tr>
<tr>
<td>60.0</td>
<td>52.1</td>
<td>449.9</td>
<td>0.3392</td>
<td>1.0000</td>
<td>0.3591</td>
<td>0.0000</td>
</tr>
<tr>
<td>80.0</td>
<td>67.29</td>
<td>430.2</td>
<td>0.4486</td>
<td>1.0000</td>
<td>0.4637</td>
<td>0.0000</td>
</tr>
<tr>
<td>100.0</td>
<td>82.1</td>
<td>409.2</td>
<td>0.5574</td>
<td>1.0000</td>
<td>1.5654</td>
<td>0.0000</td>
</tr>
<tr>
<td>120.0</td>
<td>96.5</td>
<td>386.2</td>
<td>0.6656</td>
<td>1.0000</td>
<td>0.6649</td>
<td>0.0000</td>
</tr>
<tr>
<td>140.0</td>
<td>110.7</td>
<td>360.2</td>
<td>0.7734</td>
<td>1.0000</td>
<td>0.7626</td>
<td>0.0000</td>
</tr>
<tr>
<td>160.0</td>
<td>124.6</td>
<td>328.8</td>
<td>0.8808</td>
<td>1.0000</td>
<td>0.8587</td>
<td>0.0000</td>
</tr>
<tr>
<td>180.0</td>
<td>138.4</td>
<td>284.6</td>
<td>0.9879</td>
<td>1.0000</td>
<td>0.9534</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
2.5 FFPU Linear Models

FFPU linear models are provided in state-space and transfer matrix form. They are formulated in a general format, suitable for investigation of the process characteristics in wide operating ranges.

2.5.1 State-space model

Equations (2.28) through (2.38) constitute a nonlinear model of the form:

\[
\begin{align*}
\dot{x} &= f(x, u) \\
y &= g(x, u)
\end{align*}
\]  

(2.50a, b)

where \(x = [x_1 \ x_2 \ x_3]^T = [E \ P \ \rho]^T\) is the state vector, \(u = [u_1 \ u_2 \ u_3]^T\) is the input vector, and \(y = [y_1 \ y_2 \ y_3]^T = [E \ P \ L]^T\) is the output vector.

The linear state-space model is obtained from a truncated Taylor series expansion of the nonlinear equations around an equilibrium point defined by \(x_e = [x_{1e} \ x_{2e} \ x_{3e}]^T\) and \(u_e = [u_{1e} \ u_{2e} \ u_{3e}]^T\) with \(y_e = [y_{1e} \ y_{2e} \ y_{3e}]^T\) as the corresponding output. The linear system matrices are given by:

\[
\begin{align*}
A &= \frac{\partial f}{\partial x}_{|_{x_e,u_e}} \\
B &= \frac{\partial f}{\partial u}_{|_{x_e,u_e}} \\
C &= \frac{\partial g}{\partial x}_{|_{x_e,u_e}} \\
D &= \frac{\partial g}{\partial u}_{|_{x_e,u_e}}
\end{align*}
\]  

(2.51a-d)

The linear model approximation is given by:

\[
\dot{x} = A\hat{x} + B\hat{u}
\]  

(2.52a)
\[
\hat{y} = C \hat{x} + D \hat{u}
\]  
(2.52b)

where \( \hat{x} = x - x_e \), \( \hat{u} = u - u_e \), and \( \hat{y} = y - y_e \) are the state, input, and output deviation vectors, respectively.

The system matrices are:

\[
A = \begin{bmatrix}
-0.1 & \frac{9}{8} (0.073 u_{2e} - 0.016) & x_{2e}^{1/8} & 0 \\
0 & -\frac{9}{8} & 0.0018 u_{2e} x_{2e}^{1/8} & 0 \\
0 & -\frac{1}{V_f} (1.1 u_{2e} - 0.19) & 0 & 0
\end{bmatrix}
\]  
(2.53a)

\[
B = \begin{bmatrix}
0 & 0.073 x_{2e}^{9/8} & 0 \\
0.9 & -0.0018 x_{2e}^{9/8} & -0.15 \\
0 & -\frac{1.1}{V_f} x_{2e} & \frac{141}{V_f}
\end{bmatrix}
\]  
(2.53b)

\[
C = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & C_{32} & C_{33}
\end{bmatrix}
\]  
(2.53c)

\[
D = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
D_{31} & D_{32} & D_{33}
\end{bmatrix}
\]  
(2.53d)

with

\[
C_{32} = 50 \left[ \frac{60 (1/v_{w} x_{2e} - v_{w}) k_{s1}}{(1 - v_{w} (k_{s1} x_{2e} + k_{s2})))^a} + \frac{T_f V_f}{a (1 + K)} (1.1 u_{2e} - 0.19) \right]
\]  
(2.54a)

\[
C_{33} = 50 \left[ v_{w} V_f - \frac{60 (k_{s1} x_{2e} + k_{s2})}{(1 - v_{w} (k_{s1} x_{2e} + k_{s2}))) x_{2e}^2} \right]
\]  
(2.54b)

\[
D_{31} = 50 \left[ \frac{T_f V_f k_{s1} k_{f1} k_{f0}}{a (1 + K)} \right]
\]  
(2.54c)

\[
D_{32} = 50 \left[ \frac{T_f V_f 1.1 K x_{2e}}{a (1 + K)} \right]
\]  
(2.55d)
\[ D_{33} = 50 \left[ \frac{T_f V_f}{a - 1 + K} \right] \]  

(2.55e)

where all constant parameter values were previously defined in Table 2.2.

### 2.5.2 Transfer matrix model

The transfer matrix model can be obtained directly from the linear state-space model in the previous section using the Laplace transform:

\[
\hat{Y} = T \hat{U} = \left[ C(sI - A)^{-1} B + D \right] \hat{U}
\]

(2.56)

where \( \hat{Y} \) and \( \hat{U} \) are the Laplace transform of the output deviation, \( \hat{y} \), and the input deviation, \( \hat{u} \), \( s \) is the Laplace complex variable, and \( T \) stands for the system transfer matrix.

The elements of the transfer function matrix are found to be:

\[
T_{11} = \left[ A_{12}B_{21}s \right] / d(s)
\]

(2.57a)

\[
T_{12} = \left[ B_{12}s^2 + (A_{12}B_{22} - A_{22}B_{12})s \right] / d(s)
\]

(2.57b)

\[
T_{13} = \left[ A_{12}B_{23}s \right] / d(s)
\]

(2.57c)

\[
T_{21} = \left[ B_{21}s^2 - A_{11}B_{21}s \right] / d(s)
\]

(2.57d)

\[
T_{22} = \left[ B_{22}s^2 - A_{11}B_{22}s \right] / d(s)
\]

(2.57e)

\[
T_{23} = \left[ B_{23}s^2 - A_{11}B_{23}s \right] / d(s)
\]

(2.57f)

\[
T_{31} = \left[ D_{31}s^3 + (B_{21}C_{32} - D_{31}(A_{11} + A_{22}))s^2 + ((A_{32}C_{33} - A_{11}C_{32})B_{21} + D_{31}(A_{11}A_{22}))s \right] / d(s)
\]

(2.57g)

\[
T_{32} = \left[ D_{32}s^3 + (B_{22}C_{32} + B_{32}C_{33} - D_{32}(A_{11} + A_{22}))s^2 + (A_{11}A_{22}D_{32} - A_{11}B_{22}C_{32} + A_{32}B_{22}C_{33} - (A_{11} + A_{22})B_{32}C_{33})s + (A_{11}A_{22}B_{22}C_{33} - A_{11}A_{32}B_{22}C_{33}) \right] / d(s)
\]

(2.57h)
\[ T_{33} = [D_{33}s^3 + (B_{23}C_{32} + B_{33}C_{33} - D_{33}(A_{11} + A_{22}))s^2 + (A_{11}A_{22}D_{33} - A_{11}B_{23}C_{32} + A_{32}B_{23}C_{33} - (A_{11} + A_{22})B_{33}C_{33})s + (A_{11}A_{22}B_{33}C_{33}) - A_{11}A_{32}B_{23}C_{33})] / d(s) \]

where

\[ d(s) = s(s - A_{11})(s - A_{22}) = s^3 - (A_{11} + A_{22})s^2 + A_{11}A_{22}s \]

### 2.6 Summary

This chapter presented a minimum number of relevant process engineering facts that must be taken into account to achieve overall process optimization under arbitrary operating scenarios. These facts included basic energy (first law of thermodynamics) and exergy (second law of thermodynamics) analysis, a synthesis of overall operation practice, and appreciation of the qualitative dynamics of a FFPU. The relevance of these facts cannot be overemphasized since, as will be shown in later chapters, they will shape the definition of the control objectives and operating policies. Very frequently, these concepts are overlooked in favor of the information that can be obtained from a dynamic model of the process; rather they point to niches for performance improvement in the FFPU.

After specifying the requirements of this dissertation, a mathematical model that remarkably captures the main process dynamics of the FFPU was explained. The model was presented in several versions including: non-linear, linear and static. In a first approximation, the model could appear deceptively simple, but after being distilled for more than two decades, the model provides unsurpassed insight into the wide-range overall behavior of the process, which is extremely difficult to obtain from, and obscured by, more complex models. The wealth of information provided by the model in its different representations complements the process engineering concepts required to achieve an effective overall control system design.
CHAPTER 3

CONTROL SYSTEM STRUCTURE

Following a software engineering approach, this chapter describes the conceptual model of the Intelligent Coordinated Control System (ICCS) up to a second level of detail. In Section 3.1, some necessary basic agent-oriented software engineering concepts directly related to the design of the overall unit control system are discussed. The first level system description of the ICCS is provided in Sections 3.2 and 3.3. Section 3.2 portrays the main concepts taken into account to shape the structure of the ICCS by describing it from several different perspectives: context physical situation, computer integrated processing, fundamental automation functions, and intelligence functions. Section 3.3 presents the resultant ICCS architecture as an open multiple-agent intelligent system and provides a functional description of its components. In Section 3.4 the second level module description is provided with respect to the ICCS-MP (ICCS Minimum Prototype), including the general remarks on the base direct control strategy. Each major module is described in detail in the following chapters. In Section 3.5 it is shown how the proposed ICCS-MP extends the concept of current coordinated control schemes at FFPUs. Finally, Section 3.6 summarizes this chapter.

3.1 Control Software Engineering

As mentioned in Chapter 1, the multidisciplinary approach amalgamating control engineering, process engineering, and software engineering concepts, for the design of the overall power unit control system is a central issue in this dissertation. In this chapter, the software engineering aspect is highlighted by taking into account that the main goal of this research, an overall power unit control system, could also be seen as the design of
a comprehensive software system to ease the incorporation of advanced operation applications \textit{(i.e.,} advanced protection, control, and automation software clients) \textit{to enhance the performance of a fossil fuel power unit.} Perhaps the most relevant issue in this regard is the interdependence between the software and control design methodologies. This interaction is seldom treated in the technical literature but for digital control system realization [Bennett 1994]. Here, it is shown how state-of-the-art artificial intelligence software engineering concepts provide a comprehensive unifying framework, in which advanced and conventional application functions can be harmoniously integrated despite the very many different technologies utilized in their development, to produce an intuitive and practical control system design that effectively deals with system complexity.

3.1.1 Need for a software engineering approach

Computer based systems have become a major driving force in modern society. All sorts of information are manipulated through software programs. Coined in 1967, software engineering refers to the discipline that provides the means \textit{(i.e.,} process models, methods, and tools) \textit{to produce high quality fault-free software on time and within budget} [Schach 1999]. Software quality may be quantitatively assessed in terms of software metrics \textit{(e.g.,} size, cost, duration, effort, accuracy, reliability, efficiency, integrity, etc.). The software life-cycle typically comprehends a series of general phases: requirements, specification, design, implementation, integration, maintenance, and retirement. The way in which these stages are arranged constitutes a software process or life-cycle model \textit{(e.g.,} waterfall, rapid prototyping, incremental, synchronize-and-stabilize, spiral, etc.).

Software engineering concepts have been used successfully for the development of many software systems. However, after more than thirty years since the inception of software engineering the flexibility offered by software is still very frequently abused. Much software is developed using the so called build-and-fix approach, without
specifications or any attempt at design. It is very common to find software systems that are extremely customized, which are very difficult to validate and maintain, and most importantly, for which it is very problematical to guarantee error-free operation. As incredibly as this could appear, there is still a shocking lack of systematization in the design of software systems. Among many others, a reason for this situation is the prolonged existence of the “early programming culture” that does not give software development the importance it deserves. The wrong believes that anybody can develop software, that software systems are easy to design, generate and test, and that software can be updated at low cost and in short time, are surprisingly still fairly popular in both the academic and industrial circles.

The impact of the application of software engineering concepts can be grasped from some quantitative facts regarding the cost of software. Most relevant figures include the cost of software with respect to the total cost of the computer-based system, the relative cost of software per phase of the life-cycle, and the relative cost of error detection and correction at every phase of the software process. These figures provide general guidelines for the application of software engineering concepts.

First, the cost of software (application and basic) may easily supersede the cost of equipment by a large amount, ranging from 50%-50% (in some telecommunication and military systems), to 90%-10% (in some very large systems), respectively. The cost of application software (modeling, algorithms, and heuristics) development is usually larger than that of the basic software (scientific libraries, operating systems, compilers, communication, and user interface) in figures that range from 60%-40% to 80%-20%, respectively. Within the application software, the cost and development effort of heuristics dominates the mathematical part (algorithms and modeling), even in the case of critical mathematical parts, by one or two orders of magnitude [Benveniste 1991].

Second, the average relative cost of software per phase of the software process show that the requirements phase accounts for 2% of total software cost, specification for 5%, design for 6%, development accounts for 12% (5% in module coding and 7% in module testing), integration for 8%, and maintenance accounts for 67% of total software
cost [Schach 1999]. As shown, maintenance is an extremely time-consuming and expensive phase of the software life cycle. A major concern of software engineering is the application of techniques and tools that lead to a reduction in maintenance costs.

Third, studies have found that between 60% and 70% of all faults detected in large-scale projects are due to specification and design errors [Boehm 1979]. If an error is made in an earlier phase, the resulting fault will propagate into the subsequent phases of the software process. The relative cost of detecting and fixing a fault increases as the software process progresses. Typical figures show a proportion of 1:3:4:10:30:200 in the relative cost of detecting and fixing a fault during the requirements, specification, design, implementation, integration, and maintenance phases, respectively [Boehm 1980]. These facts show the convenience of detecting and fixing faults as early as possible, and the relevance of the specification and design phases.

All issues mentioned above directly apply to the development of software for computer-based monitoring and control systems (control software systems), where their relevance is increased to a new dimension when involved in critical applications such as military defense, flight control, health care, power supply, etc. The development of real-time high-risk control software systems poses more challenges than the data processing systems for scientific and office applications [Leveson 1990]. Real-time requirements may include: continuous operation, execution depending on external events or at fixed frequency rates, direct and immediate data entering from outside the system, immediate data processing for process control, direct effect on the controlled process, low rate of printed information, and distributed processing. Consequently, control software systems may easily become very complex systems. Furthermore, the integration of new advanced applications to achieve more versatile and efficient operation of the system under control, and to exploit more fully the digital hardware capabilities, will again notably increase the complexity of any control software system. Every addition or change may become a potential risk and should be thoroughly tested at all the stages of the software process.

Although it has been recognized internationally that the development of high-quality control software systems requires dedicated methods and highly trained personnel
[ISO 9000-3 1991], it should always be kept in mind that adherence to standards is not intended to achieve 100% software quality per se, but to reduce the risk of poor quality software. Also, software engineering procedures do not provide the creativity needed for the invention of the essential philosophy and heuristics in the initial design of a large control system. Clearly, the initial design of large control software systems still remains an art, that heavily relays on the aptitude and experience of the designer(s). The application of software engineering techniques may systematically assist the art of control system design to achieve completeness and to prevent conceptual errors through successive refinement. Software engineering may help to tackle the complexity of the control system design in a systematic way.

The matter of crux is that since a control system design will eventually be translated into a software system, it is better to assist its design with software engineering concepts in mind from the very beginning. In turn, this approach will ease the transition from the requirements, specification, and design phases to the development phases, reduce time and cost of development, and minimize software errors at later stages in the software process. It is firmly believed that when properly applied, the discipline of software engineering will certainly produce clean-cut overall control system designs, where the benefits of the advanced operation applications can be more easily recognized, and successively improved.

### 3.1.2 Control software problems and challenges

Around thirty years ago, the early control software systems were developed as direct translations of their analog counterparts. All system functions were performed by a single program, with all system data available to all functions. From the beginning it was expected to obtain systems which were easier to duplicate, and to upgrade with more sophisticated control strategies. Nevertheless, the high cost and tight constraints on processing power and memory capacity, together with a lack of software development
tools, yielded highly customized software structures with tight couplings between the timing and control functions, which were exceptionally difficult to modify.

Through the years, and with the availability of more powerful hardware and software, the application of computer-based control systems spread out. Control software systems substantially increased in size and complexity. It was then required to design the systems at higher levels of abstraction and to built them by larger development teams. System structures become modular, so that different modules can be developed by different groups. All sort of modules can be created. On the one hand, dependencies between any two modules and a large number of modules yield intricate patterns of communication within the system. As a consequence, the integration of modules in a large control system can take as long as their development. On the other hand, the availability of high level programming languages, specially tailored to the implementation of control systems, although has certainly benefited the control engineering community, has not had the expected success. Reasons are that the use of high level languages made control over timing far more difficult than without them, and that the dependencies among modules are a far greater problem than the actual programming of the required algorithms, in fact, the problem has never been the coding of the algorithms. Another concern arises with the need to use multiple processors to meet the processing demands of the control system. Concurrent processing raises new problems because parallel processing has not been available before, and the development of procedures for parallel execution requires the availability to assess behavior and correctness. Unfortunately, the presence of multiple processors, even as redundant units for increased reliability, leads to a sharp increase in system complexity.

The traditional approach to control system design presents two major problems: data consistency and timing between the various components of the system. Normally, the relation between two software modules is much more complicated since not only parameters and functions results are shared, but also global variables. In order for the various modules to correctly interpret the data they operate on it is necessary that the access to the data is well controlled. In many cases this control is assured through a
procedure-call interface (parameter passing). In others, explicit control mechanisms need to be introduced to ensure consistency. This is particularly true in systems where several modules are active at the same time, either through interleaved execution in a single processor or on truly concurrent machines. In complex systems a large number of interactions between modules need to be managed in terms of both the data that is transferred and control over the transfer itself. The potentially very large number of interfaces to be managed is one of the major reasons for the difficulties experienced in designing large complex systems. Besides making the design process difficult, the built-in connections between modules also cause the resulting system to be static. For flexibility at run-time, which is needed to reconfigure a system, it is vital that the communication between modules is not fixed during the design but can be established as the need arises. As another fundamental problem, the major issue with timing is the instability that could occur if the computational delay between measuring and controlling the process is too large. This is a physical constraint that have to be met by the control system. Other less critical timing constraints arise when human operators are part of the control mechanism. These timing constraints cause problems for system development only if the processing power is a critical issue. If insufficient computing power is available, the only solution is to split the computation into parts that can be executed in parallel, and add processors in a dedicated configuration. Determining the ability of a system to actually meet its deadlines under all circumstances is very difficult and is currently beyond the state-of-the-art in multiprocessor systems. The effect of the development of complex control systems is to simplify the problem and design for the worst case situation, in which all processes will have to meet their deadlines concurrently.

Much effort has been spent in solving the above mentioned problems. Nearly all the approaches have concentrated on providing more powerful tools for the implementation of the specifications in a function-oriented approach. Typical examples of these efforts are high level programming languages and real-time operating systems. In parallel to these efforts for developing better tools for implementing systems, extensive
work has been done in the area of providing support for the development process itself. The so called software development environments aim at providing mechanisms through which the complexity of the design can be managed. These systems are generally very complex themselves, and have not demonstrated to reduce the design time or achieve higher system quality. From recent publications it is clear that increasingly, researchers are looking for ways to achieve the prevention of complexity, although the majority of effort is still directed to developing better tools. Alternative efforts are aiming at architectures that simplify the implementation of the desired systems. To prevent unnecessary complexity, neither the design method nor the selected architecture should increase the system’s complexity beyond the intrinsic complexity of the specification to be realized by the system.

3.1.3 Agent-based design

The development of control software systems, considered as real-time systems and high-risk in many occasions, requires the utilization of proper development methods, different from those required by scientific and business applications. The development of real-time control systems was initially carried out with methods originated in the non-real-time applications with adequate modifications. Some of these methods include real-time structured analysis and design [Ward and Mellor 1985, Gomma 1984], and real-time object-oriented design [Neidert 1993].

Agent based design has been proposed as a new computation paradigm to design and implement high quality software for large complex open and distributed applications [Sycara 1998]. An open application (system) is one in which the structure of the system itself is capable of changing dynamically. The information sources, communication links, and components could appear and disappear arbitrarily and unexpectedly. Also, the components may not be known in advance, change over time, and be highly heterogeneous. The components may be implemented by different people, at different
times, and with different tools and techniques. Multiagent systems tackle complexity through modularity and high level abstraction by developing a number of functionally specific components (agents) that are specialized at solving a particular problem aspect. Each agent uses the most appropriate paradigm for solving its particular problem and can be coordinated with another agents to properly manage interdependencies. Agent based methodologies have been proposed to significantly enhance our ability to model, design and built complex distributed software systems [Jennings 1995].

Even if there is no general consensus on the definition of an agent, an useful working definition states that an agent is an encapsulated software system situated in some environment, capable of flexible autonomous action in that environment in order to meet its design objectives [Wooldridge 1997]. More simply, an agent is software that can carry out information-related tasks without on-going human supervision [Knapik and Johnson 1998]. A Multiple Agent System (MAS) can be defined as a loosely coupled network of problem solvers (agents) that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver [Durfee and Lesser 1989]. The problem solvers are autonomous and heterogeneous in nature.

The main characteristics of MASs are: 1) Each agent has incomplete information or capability for solving the overall problem, and thus, has a limited viewpoint. 2) there is no system global control. 3) Data is decentralized, and 4) Computation is asynchronous. The motivation for the use of MASs is due to their ability to: 1) Solve problem that are too large for a single centralized agent. 2) Allow the interoperation of multiple existing systems to keep pace with changing needs. 3) Provide solutions that efficiently use information sources that are spatially distributed. 4) Provide solutions where expertise is distributed. 5) Enhance performance along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reusability.

Although MASs provide many potential advantages, they also present many difficult challenges for their design and implementation. The first essential problem consist of the formulation, description, decomposition and allocation of the overall
problem, followed by the synthesis of a group of intelligent agents (main concern in this dissertation). Other problems include enabling agents to communicate and interact, enabling agents to reason about the actions of other agents, ensuring that agents act coherently in decision making, recognizing and reconciling disparate viewpoints and conflicting intentions among agents to be coordinated, engineering practical systems, etc.

The design of MASs architectures, particularly layered architectures, is an area of considerable current research. An organization provides a framework for agent interactions through the definition of roles, behavior expectations, and authority relations. Organizations are, in general, conceptualized in terms of their structure, that is, the pattern of information and control relations that exist among agents and the distribution of problem solving capabilities among them. The structure should also indicate the connectivity information to the agents so they can distribute sub-problems to competent agents. In open dynamic systems the issue of organizational adaptivity is crucial. Organizations that can adapt to changing circumstances by altering the pattern of interactions among the different constituent agents have the potential to achieve coherence. Agents may dynamically find their collaborators based on the requirements of the task and on which agents are part of the organization at any given time, thus adaptively forming teams on demand [Decker, et al. 1996]. Early work has dealt with groups of agents pursuing common goals [Lesser 1991].

The MAS paradigm has already been used to implement large software structures for monitoring and control systems. First MAS applications appeared in the mid 1980s [Jennings, et al. 1998]. In [Durfee 1987, Durfee and Lesser 1989] a set of geographically distributed agents monitor vehicles to track vehicle movements in a global area. In [Parunak 1987] a MASs is used to effectively manage the production process in a manufacturing enterprise where each factory is represented as an agent. In [Ljunberg and Lucas 1992] a sophisticated agent-based air-traffic control system where aircrafts and air traffic control systems are represented by agents. Here the agent paradigm provides a useful and natural way of modeling real-world autonomous components In [Schwuttke and Quan 1993] a MAS is used to monitor and diagnose faults in spacecraft control. In
[Boasson 1993] the MAS paradigm is explained for a radar tracking system. The best-known MAS applications in process control are those in [Jennings, et al. 1995]. MASs were developed for power transmission management and particle accelerator control using the ARCHON software development platform for MASs. In [Velasco, et al. 1996] a MAS was designed for distributed control of industrial processes and applied to a fossil fuel power plant to improve the heat rate.

### 3.1.4 Agent-oriented technologies and methodologies

Agent technology has received a great deal of attention in the last few years, driven mainly for internet applications. There is different developed agent theories, languages, architectures, and successful applications. Nevertheless, very little work has been done on the development of methodologies to specify and develop applications using agent technology. The role of agent-oriented methodologies is to assist in all the phases of the life-cycle of an agent-based application. Researchers on agent-oriented methodologies have followed the approach of extending existing methodologies to include the relevant aspects of the agents. Extensions have been mainly proposed to object-oriented (OO) methodologies and knowledge engineering (KE) methodologies.

Similarities with the object-oriented paradigm consider agents as active objects, that is, objects with a mental state. Both paradigms use message passing for communicating and can use inheritance and aggregation in defining the system architecture. The main difference is the constrained type of messages in the agent-oriented paradigm and the definition of a state in the agent based on its beliefs, desires, intentions, commitments, etc. Agents are not simple objects. Aspects do not addressed by object-oriented methodologies include the types of messages used by agents. Agents model messages as speech-acts and use complex protocols to negotiate. Also, agents analyze these messages and can decide whether to execute the requested action. Another difference consists in that agents can be characterized by their mental state, and object-
oriented methodologies do not define techniques for modeling how the agents carry out their inferences, their planning process, etc. Finally, agents are characterized by their social dimension. Procedures for modeling these social relationships between agents have to be defined. Agent oriented methodologies that extend object-oriented methodologies include:

- **Agent-oriented analysis and design.** Proposes three models for analyzing an agent system: the agent model, that contains the agent and their internal structure; the organization model, that describes the relationships between agents; and the cooperation model, that describes the interactions between agents.

- **Agent modeling techniques for systems of BDI (Belief, Desire, and Intention) agents.** This method considers two main levels for modeling BDI agents: the external viewpoint, that decomposes the system into agents and their interactions using the agent model and the interaction model; and the internal viewpoint, that carries out the modeling of each BDI agent using three models: the belief model, the goal model, and the plan model.

- **Multi-agent scenario-based (MASB) method.** Intended for MAS design in the field of cooperative work. Considers two phases: analysis and design. Activities included in the analysis phase are: scenario description, role functional description, data and world conceptual modeling, and system-user interaction modeling. Activities included in the design phase are: MAS architecture and scenario description, object modeling, agent modeling, conversation modeling, and system design overall validation.

Knowledge engineering methodologies provide the techniques for modeling agent knowledge. The definition of the knowledge of an agent can be considered as a knowledge acquisition process, and only this aspect is considered in knowledge-engineering-based agent methodologies. Designing a MAS shows most of the same problems faced by KE methodologies: knowledge acquisition, modeling, and reuse. KE conceives centralized knowledge based systems, without addressing the distributed or social aspects of agents, or their reflective and goal-oriented attitudes. MAS modeling
solutions proposed in this direction include extensions to CommonKADS, which can be seen as an European standard for knowledge modeling [Iglesias, et al. 1998]:

- The CoMoMAS methodology models a MAS through the following models: Agent model, Expertise model, Task model, Cooperation model, System model, and Design model.
- The MAS-CommonKADS methodology extends the models defined in CommonKADS using techniques from object-oriented methodologies, and from protocol engineering to describe the agent protocols. This methodology starts with an informal conceptualization phase to collect user requirements and to obtain a first description of the system from the user point of view. This methodology defines the following models: Agent model, task model, expertise model, coordination model, organization model, communication model, and design model. This methodology has been applied in several research projects in different fields.

Some agent developers have successfully developed and applied MAS to practical problems. Although not declared as a MAS methodology, they constitute an important approach for the design of MAS that should also be considered, since they have provided general guidelines for MAS development. That is the case of the ARCHON development environment for MAS, which proposed a methodology for analyzing and designing MAS. The analysis combines a top-down approach that identifies the system goals, the main tasks and their decomposition, with a bottom-up approach that allows the reuse of preexisting systems constraining the top-down path. The design is subdivided into agent community design and agent design. The agent community design defines the agent granularity and the role of each agent. The agent design encodes the skills for each agent.

Even if there are no standard definitions for an agent, an agent architecture, or an agent language, the methodologies reviewed here show that there exist a conceptual level for analyzing agent-based systems, no matter what the agent theory, architecture or language supports them. This conceptual level should describe:

- Agent models: the characteristics of each agent should be described, including skills (sensors and effectors), reasoning capabilities and task.
• Group/society models: the relationships and interactions between the agents.

The lack of standard agent architectures and agent programming languages is currently the main problem for MAS model building and implementation. Also, since there is no standard architecture, the design of the agents needs to be customized to each agent architecture. Also, some methodologies select the agent architecture during the analysis, while others consider that it is a design issue, and the agent architecture should be selected depending on the requirements of the analysis.

3.2 Design of ICCS Architecture

From a software engineering perspective, the development of the ICCS in this dissertation may be related to a software process life-cycle. The contents of Chapters 1 and 2 constitute a general specification of requirements. This chapter proposes the design of the ICCS as a multiagent system, describes the conceptual model of the ICCS, and specifies a minimum prototype (ICCS-MP) for feasibility demonstration purposes. Chapters 4, 5, and 6 undertake the implementation and validation (module testing) of the ICCS-MP main components. Finally, Chapter 7 describes the integration and validation (system testing) of the ICCS-MP.

A brief historical note is provided to explain the approach undertaken in the design of the ICCS architecture. The conceptual model of the ICCS in this dissertation constitutes the most recent of an evolving series of control systems structures that the author has been engaged in for automation of power plants and processes. This effort started several years ago motivated by the development of a structured control software interpreter for batch process automation of the water demineralization plant in a FFPU [Mundo 1995]. The interpreter implemented a general functional structure comprising modules for logic signals handling, analog signals handling, protections, regulatory control, logic sequences, recipe handling, and operation interface. In a parallel development, the search of a general control structure for FFPUs under automatic
generation control through reconfigurable schemes [Tevera 1995] pointed out the need for a structure to make emphasis on the supervisory functions to attain higher levels of automatic operation. Then, the basic interpreter structure was modified to automate a whole power unit. The recipe handling features were replaced by a more general supervision module to incorporate the set-point generation functions and non-linear characterizations necessary for operation of an actual power plant. The new structure was applied to develop the control software system for small size (1 MW) and medium size (60 MW) turbogas units [Ramirez 1996, Garcia 1997]. Soon after, it was realized that even this modified structure should be additionally enhanced to achieve greater versatility. Artificial intelligence techniques were suggested to achieve self-governing characteristics [Garcia and Garduno 1998]. At the present time, a general control structure for FFPU's that incorporates the agent software notion as the fundamental design premise for implementing intelligent hybrid control systems is proposed.

The design of the ICCS as a multiagent system incorporates several interwoven concepts. To appreciate that, in what follows, the definition of the multiagent system structure is described as a sequential process that incorporates those concepts step-by-step. In this way, the ICCS architecture is described from various dimensions: physical context situation, industrial hierarchical structuring, extended process automation, and intelligence functional grouping.

3.2.1 Context of the ICCS

Being part of a power system, a FFPU demands on its control system the capability to be incorporated into a larger control system. Thus it is very important for the ICCS to provide the means to be coordinated with all other power system control elements. Pyramidal automation models for power systems have been around for several years [Schweppe and Mitter 1972, Malik, et al. 1980]. These models promote the implementation of large-scale vertical hierarchical automation structures consisting of
several levels, with the power units at the bottom level. Nevertheless, the current trend toward market liberalization of the electric sector predicts substantial changes in the management and arrangement of the upper levels in these structures [Ilic and Liu 1996].

Since power plant physics are not subject to these trends, it is expected that power units used for load-frequency control in deregulated power systems will be commanded as self-supported energy production cells through the sole specification of unit load commands. That is, the power system restructuring trend will put even more emphasis on the role of power units as power system actuators, in the way that control valves are used to regulate process variables, to control power generation.

Thus, for the aims of this dissertation, it will be assumed that a power unit is commanded through a single unit load demand profile, regardless of being located either at the bottom level of a larger hierarchical power system or as part of a generation-only utility in a deregulated power system (Figure 3.1). Also, optimal power unit operation is considered a local affair; procured at the sole discretion of the unit operators and motivated by survival under competition and profit achievement.

Figure 3.1 Context interaction of a power unit.
3.2.2 Industrial hierarchical structure

In essence, the architecture of the ICCS is formulated as a hierarchical system taking into account the concepts from industrial process automation engineering and basic principles for the implementation of intelligent systems. In brief, the architecture of ICCS comprehends the bottom levels of a general industrial automation hierarchical model, where each level performs the basic functions of an intelligent system in different degrees. The hierarchical levels considered by ICCS are the direct control level and the supervisory control level (Figure 3.2), which correspond to levels 1 and 2, respectively, in the reference model for computer integrated manufacturing (CIM) for a continuous industrial process in [Williams 1989].

According to this reference model, the direct and supervisory levels concentrate in performing control computation and control enforcement tasks, while the upper levels concentrate in production scheduling and management information processing tasks.

Generic duties for the direct control level include:

• Maintain direct control of the plant under its cognizance.
• Detect and respond to any emergency condition which may exist in this plant.
• Collect information on unit production, raw material and energy use and transmit to higher levels
• Service the operator’s man-machine interface.

Generic duties of the supervisory control level include:

• Respond to any emergency condition which may exist under its cognizance.
• Optimize the operation under its control within the limits of established production schedule.
• Carry out all established process operational schemes or operating practices in connection with these processes.
• Collect and maintain data queues of production, inventory, and row materials and energy usage for units under its supervision.
• Service the supervisor’s man-machine interface.
management data presentation

management information

sales orders

operational and production supervision

productionscheduling and operational management

communication with other areas

supervisor's console

intra-area coordination

communication with other supervisory systems

supervisor's console

supervisory control

communication with other control systems

operator's console

direct control

communication with other areas

dedicated controllers

PROCESS

Figure 3.2 Hierarchical control structure.
3.2.3 Fundamental functional structure

The fundamental functional structure of the ICCS is based on a rather general structure model for industrial batch-process automation. This structure model has been applied successfully for batch process automation [Rosenof and Ghosh 1987] and for building highly flexible batch process control software [Mundo 1995]. In this model all functions performed by a control system are grouped into seven basic functional groups, as shown in Figure 3.3, where the recipe management functional group has been replaced by what was named the supervision functional group, since from the batch process control point of view, a power unit produces only one product: electric power.

**Logic/analog input/output signal handling functions:** Constitute the interface between the control functions and the process. They are responsible for entering and sending contact signals, as well as continuous signals, from and to the process instrumentation at regular intervals or under demand.

**Interlocks and protection functions:** Monitor critical variables to avoid the process entering unsafe operating regions. They depend heavily on the physical characteristics of the plant and on the safety requirements of the process.

**Sequence logic functions:** Specify how and in what order the plant operations should be executed. They allow the transition between the various operating states. Enable and disable the continuous control functions as required.

**Regulatory control functions:** Evaluate control algorithms for driving the continuously varying signals in the process.

**Supervision functions:** Provide the set-point values and non-linear characterizations needed at different operating states and to deal with non-linear process dynamics.

**Operator interface functions:** Allow for interaction between operator and controller. They must be able to provide the operator with relevant information about the plant behavior. Must also provide a means to modify control parameters and strategies.
At least implicitly, all functional groups are currently performed in many industrial control systems. The ICCS embraces all functional groups, except the signal handling functions due to their high dependence on the implementation hardware.

Figure 3.3 Batch process control architecture.
3.2.4 Intelligence functional structure

The fundamental functional structure in the previous section has demonstrated its value to implement highly structured control software systems [Ramirez 1996, Garcia 1997]. Extension of the capabilities of this structure was suggested to use artificial intelligence techniques [Garcia and Garduno 1998]. The two most prominent principles to implement intelligent systems are used as guidelines to expand the scope of the fundamental structure in a systematic way.

In [Albus 1991] four basic intelligence functions are identified as primary components of an intelligent system: sensory processing, world modeling, value judgment, and behavior generation.

- Sensory processing implements perception functions and compares observations with expectations to compute physical and dynamic attributes of the process and its environment.
- World modeling obtains an estimate of the state of the world (process, environment, and control system) based on information from sensory processing, to generate expectations and predictions to be used by the other functions.
- Value judgment evaluates and grades the system behavior computing costs, risks, and benefits of observed situations and of planned activities.
- Behavior generation selects goals and plans, executes tasks, and monitors execution of plans and modifies them when required.

Figure 3.4 shows a two-level hierarchical structure with all four intelligence functions in each level.

In the ICCS, the two-level hierarchical structure is also considered, but different emphasis is done on the intelligence functions at each control level, in accordance to the principle of increasing precision with decreasing intelligence [Saridis 1989], and to the generic functions of the CIM model. At the direct control level, more precise less intelligent and closer to the power unit, emphasis is done on rapid accurate computing over data information for plant reactive control. This level comprises the functions done
by current direct-digital control systems: logic sequences, control algorithms, and protection routines. At the supervisory control level, less precise more intelligent, emphasis is done on computations over knowledge information for plant proactive control. This level comprises some of the functions currently done by the operator and process control engineer, such as performance monitoring, controller tuning, production

Figure 3.4 Intelligence functions in two-level system.
optimization, and set-point generation. Figure 3.5 shows the resulting intelligence functions as considered in the ICCS.
3.3 ICCS General Functional Specification

3.3.1 Architecture

The ICCS is proposed to realize a multi-agent system for which a multidisciplinary approach, that amalgamates control, process, and software engineering concepts, has been followed in its design. The ICCS’s system goals are identified using power plant process engineering concepts, and intelligent control systems engineering concepts are used to identify main tasks and to functionally decompose the system. A software engineering agency concept is used to identify and group agents according to their knowledge and purpose interactions.

The proposed ICCS organization is an open superset of functionally grouped agent clusters in a two-level hierarchical system (Figure 3.6). The upper level, which is mainly characterized for knowledge-driven processes, performs the supervisory functions needed to provide self-governing operation characteristics, while the lower level, which is mainly characterized for data-driven processes, performs the fast reactive behavior functions necessary for hybrid (discrete and analog) real-time control and protection. The term organization is preferred to emphasize the soft nature of the system structure over a rigid inflexible architecture.

Agents are loosely clustered taking the intelligence functions (Section 3.2.4) as guidelines. A generic control cluster takes account of the sequence control, regulatory control, protection, and input-output handling agents. A self-awareness cluster is introduced to group the system operating state determination, fault diagnosis, and test assistance agents. The world modeling cluster comprehends the learning, model building, and adaptation agents. The value judgment cluster comprehends the on-line performance monitoring, control tuning, and reconfiguration agents. The memory cluster is introduced to include the data (sensory) and knowledge processing agents, as well as the system knowledge and data bases agents. The behavior generation cluster groups the process optimization, sequence generation, and set-point generation functions.
Figure 3.6  ICCS multiagent organization.
Agent clustering is introduced to simplify the representation and to indicate that the agents in a cluster use closely related system knowledge or data, and have mutual commitments and beliefs. In reality, all agents may coexist as parallel processes with random access to system information. As required for an open system, the ICCS exhibits organizational adaptability mediated by the supervisory execution manager agent, and the direct execution manager agent. In principle, the ICCS organization can adapt to changing circumstances by activating or deactivating agents, incorporating new agents or dismissing old agents, or modifying the pattern of interactions among the current agents in the organization. The ICCS agents should dynamically find their collaborators based on the system requirements at hand and on which agents are present in the organization at any given time. Clusters should be formed adaptively as required.

3.3.2 Functional specification

The system functional decomposition into agents in Figure 3.6 is not exhaustive in any way, it only shows what is considered a basic set of tasks that should be taken into account to achieve a more general design toward truly intelligent control systems, and how they should be organized. In the spirit of an open system, this set of tasks may be augmented, or decreased, as required by the application at hand.

*Input/output signal handling.* Constitute the interface between the control system and the plant instrumentation. Basically, it is responsible for entering and sending contact signals, as well as continuous signals, from and to the process instrumentation at regular intervals or under demand. In a more advanced application it should also implement the dialogs with new intelligent instrumentation. Also can take care of simulating a virtual environment for the entire system by halting actual inputs and imposing arbitrary input values.

*Protection and interlocking.* Monitor critical variables to prevent the process entering unsafe operating regions, or shutdown total or partial process when already in
unsafe conditions. These functions depend heavily on the physical characteristics, equipment configuration, and protection requirements at different operation stages of the process.

*Continuous regulatory control.* Evaluates control algorithms for driving the continuously varying signals in the process according to predefined references.

*Sequence control.* Allows the transition between the various operating states of the process. Enables/disables the continuous control functions as required.

*Operating state determination.* Evaluates key signals to declare the operating state of the FFPU. This information is to be used by other functions in decision-making and evaluation of permissive conditions.

*Fault diagnosis.* Identifies features of faults before occurring, and determines the causes when already occurred. Generates information for fault accommodation.

*Test assistance.* Sets and verifies all necessary conditions to perform operation tests from a given catalog of tests.

*Process optimization.* Determines the optimal operating conditions by solving optimization problems based on physical principles.

*Operation sequence scheduling.* Performs time-sequenced decisions for automatic plant operation, for instance, unit scheduling of power generation based on AGC demands and physical conditions of equipment at the FFPU.

*Set-point scheduling.* Generates set-points for the continuous control functions, and limits and threshold values for the sequence control and protection functions according to the different operative stages and optimization routines.

*Performance monitoring.* Evaluates behavior of process under control to generate meaningful performance indications for adaptation and optimization.

*Control tuning.* Based on performance values decides whether the current control configuration needs to be tuned or not. Performs tuning by updating parameters and knowledge of the direct control scheme if enabled by the human supervisor.

*Control reconfiguration.* Based on performance values decides whether a change on the current control configuration should be done or not. Suggest a different control
strategy from a given catalog and sets conditions for switching if enabled by human supervisor.

Learning. Allows the supervisor to build and modify the knowledge and data bases for inferences and decision-making required at both the supervisory and direct control levels based on observations of the input-output behavior of both the process and the control system itself.

Model building. On line modeling feature to account for plant and environment changes, provides information to be used by the adaptation mechanisms.

Adaptation. In a broad sense provides the mechanisms to deal with changes in the plant and its environment at the supervisory level, such as updating the operating sequences and nonlinear characterizations required for wide range operation.

What is relevant in the ICCS structure is the decomposition of the supervision function shown in Figure 3.3 into several other tasks, in the form of agents, to provide the control system with the capability to satisfy increasing performance demands keeping the complexity of the system within manageable terms. Also, the software agency concept provides the necessary mindset to integrate very dissimilar technologies in a systematic and harmonious way to achieve a practical and effective system; by making use of the best characteristics each technology has to offer.

3.3.3 Physical scope of ICCS

As pointed out in Chapter 1, the ICCS is intended to extend the capabilities of current coordinated control schemes at FFPUs, to attain wide range cyclic operation with optimization of multiple operation objectives (e.g., load tracking, duty life, heat rate, and pollutant emissions), during normal operating state. To this aim, the ICCS may be seen as an open system in which direct control functions can be incorporated into an overall unit control strategy, that is, the ICCS can modify its control scope over the unit by absorbing into the coordination strategy as many direct control functions as required to satisfy the
operating objectives under consideration. The control functions can be either some of the generic control loops already existing in all power plants or new ones proposed to fit particular requirements of any given plant. What matters here is the capacity (openness) of the control system to modify its scope in a systematic way. This openness characteristic is also carried out into the supervisory level, where new knowledge-processing functions, currently not available at FFPUs, can also be incorporated arbitrarily. These high-level functions are needed to exhibit effective and versatile self-governing operation characteristics, as those performed by dexterous operators and process engineers. Because of its relevance, a more detailed discussion of this issue is separately provided.

Figure 3.7 provides an example to illustrate the openness characteristic of the ICCS. In this case, the control functions comprehended by the ICCS would be the current coordinated control strategy, the steam flow and turbine speed/load control, the fuel/air flow and combustion rate control, the feedwater flow and drum level control, and the water spray flow and steam temperature control. According to the concepts presented in Chapter 2, a single coordination strategy embracing these control functions may provide the means for optimization of load tracking, duty life, heat rate, and pollution emissions. Also shown in the same figure, a supervisory function provides the set-points to the direct control loops considered. Obviously, many other control loops and supervisory functions can still be included in the ICCS.
Figure 3.7 Physical scope of ICCS.
3.4 ICCS Minimum Prototype

The development of a comprehensive ICCS for an actual power plant is a formidable task well beyond the scope of this dissertation; its magnitude and complexity are better suited for a multidisciplinary team of specialists. Instead, a prototype, with a minimum set of functions, that preserves the essential objective of the ICCS is proposed for development with the aim of demonstrating the feasibility of the proposed paradigm. The prototype is called the ICCS Minimum Prototype (ICCS-MP).

The ICCS-MP synthesizes an answer to all main issues presented in this and the previous chapters. Current electric utilities face operation scenarios characterized by a multiplicity of everyday tighter and changing requirements, e.g., load following cyclic operation, plant life extension, heat-rate improvement, reduction of pollutant emissions, etc. An integral automation approach for operation and control to keep plants running profitable is of paramount importance for the survival of any electric utility. The operation philosophy of the plant is defined and better reflected by the continuous control functions; the overall control strategy comprehends the core continuous control functions needed to drive the unit components as a single entity in its intended main productive state. Also, since most power units are part of large and complex power systems, an integral automation approach must take into account the global power system needs to achieve high levels of coordination and maneuverability for large scale optimal and robust operation.

Essentially, the ICCS-MP provides the means to achieve optimized wide-range cyclic operation by being able to follow any given unit load demand profile, which is issued by upper level economic dispatch and unit commitment systems, and to optimally accommodate an arbitrary number of generally conflicting operating objectives, which are determined by the current operating scenario and specified and prioritized by the plant operators, during normal load operating state.
3.4.1 Functional scope of minimum prototype

Developed through several stages, the ICCS-MP finally implements a two-level hierarchical intelligent hybrid multi-agent coordinated control system (Figure 3.8). The supervisory functions include optimization and set-point scheduling, learning and control tuning, and performance and state monitoring. The direct level consist of the regulatory control functions. All these functions were defined in Section 3.3.
3.4.2 Physical scope of minimum prototype

As described in the previous chapter, in a drum-type FFPU the essential overall dynamics may be described in terms of the major inputs (fuel and air flows into the furnace, steam flow into the turbine, feedwater flow into the boiler, and spray flows at the superheater and reheater) and outputs (electric power, steam throttle pressure, drum water level, superheater outlet temperature, and reheater outlet temperature). Electric power and steam pressure are tightly coupled and are affected heavily by the fuel and air flows and the steam flow. Feedwater flow slightly affects power and pressure, but greatly affects the drum level, which in turn is considerably affected by the fuel and steam flows. Similarly, the spray flows have a minor effect on power and pressure, but greatly affect the heaters outlet temperatures, which are heavily influenced by the fuel flow. Consequently, fuel and steam flow may be used to drive the unit to the desired values of power and pressure, this will disturb the drum water level and heaters outlet temperatures, which may then be manipulated with the feedwater and spray flows, respectively. The interactions among fuel, steam, and feedwater flows as inputs, and power, pressure and water level as outputs, greatly affect wide-range operation. Spray flows and temperatures can be considered for further improvement. The ICCS-MP concentrates in the former situation.

The ICCS-MP control strategy builds upon current coordinated control schemes, which typically account for simultaneous control of electric power output and main steam pressure. The scope of coordination over the internal processes of the power unit is extended to include the control of the drum water level to achieve balanced overall plant operation at all loads.

The physical scope of the ICCS-MP is shown in Figure 3.9. From a power system perspective, the power unit is commanded through a unit load demand pattern issued by upper level economic dispatch and unit commitment systems, and by an arbitrary number of operating objectives specified and prioritized by the plant operators. The ICCS-MP provides as control signals the demands on position to the valve actuators that control the
mass flow rates of fuel ($u_1$ in pu), steam to the turbine ($u_2$ in pu), and feedwater to the drum ($u_3$ in pu). The ICCS-MP receives from the process the electric power ($E$ in MW), drum steam pressure ($P$ in kg/cm$^2$), and drum water level deviation ($L$ in m).

![Diagram](image)

Figure 3.9 Physical scope of ICCS-MP.
3.4.3 Control structure

The control structure of the ICCS-MP is determined by the optimization and regulatory control functions. A rather general hybrid feedforward and feedback control structure, to be explained later, was chosen to embed the regulatory control functions. Then, the ICCS-MP is developed in three main modules: reference governor, feedforward control processor, and feedback control processor (Figure 3.10). The performance and state monitoring, and learning and control tuning functions are implicitly included in the main modules.

The set-points for electric power, steam pressure, and drum water level, \(y_d = [E_d, P_d, L_d]\), are calculated at the reference governor module. The feedforward and feedback modules provide the feedforward control signal, \(u_{ff} = [u_{1ff}, u_{2ff}, u_{3ff}]\), and the feedback control signal, \(u_{fb} = [u_{1fb}, u_{2fb}, u_{3fb}]\), respectively. Both control signal vectors are added to provide the demands, \(u = [u_1, u_2, u_3]\), to the fuel, steam, and feedwater control valves, respectively. Measured variables, \(y = [E, P, L]\), include the electric power, main steam pressure, and drum water level deviation, respectively.
From a global perspective, the reference governor specifies and coordinates the desired response of the power unit through the set-point trajectories. The feedforward and feedback paths implement a two-degrees-of-freedom nonlinear multivariable controller. The feedforward control path should provide the main contribution to the control valve demands to achieve wide-range operation. The role of the feedback control path is now complementary; it supplies the control signal component necessary for regulation and disturbance rejection in small neighborhoods around the commanded trajectories.

In Figure 3.11, the main modules blocks are expanded up to a second level of detail to show the location of the software agents with respect to the control structure, as well as the techniques involved in their development.

3.4.4 Feedforward/feedback strategy

The multivariable two-degrees-of-freedom feedforward-feedback control strategy in the ICCS-MP is proposed as an extension of a general linear single-input-single-output
feedback control loop to the nonlinear multivariable case that must achieve wide-range operation. The rationale is as follows.

In an industrial multivariable process, a SISO feedback control loop is typically built as in Figure 3.12, where \( r, u, \) and \( y \) are the reference, control, and output signals, respectively, and \( G(s) \), and \( G_c(s) \) are the process and controller transfer functions, respectively; \( s \) being the Laplace variable. In this framework, the disturbance signal, \( w \), and the disturbance transfer function, \( G_i(s) \), account for the interaction effect of any other process input through the process coupled dynamics, which is a distinctive characteristic of multivariable processes.

The transfer function of the fundamental control loop is:

\[
Y(s) = \left[ 1 + G(s)G_c(s) \right]^{-1} [G(s)G_c(s)R(s) + G_c(s)W(s)]
\]  

(3.1)

where \( R(s), W(s), \) and \( Y(s) \) are the Laplace transform of \( r, w, \) and \( y, \) respectively. From (3.1), it can be seen that this control scheme cannot provide the means to achieve perfect reference tracking and interaction disturbance rejection, only through the design of \( G_c(s) \). To that aim, the fundamental control loop scheme can be extended through reference feedforward and disturbance feedforward control, as shown in Figure 3.13, to obtain the transfer function:

\[
Y = \left[ 1 + GG_c \right]^{-1} \left[ (GG_c + GG_r)R + (G_i + GG_d)W \right]
\]  

(3.2)
where, omitting the dependency on the Laplace variable for brevity, $G_r$ is the reference feedforward control transfer function, and $G_d$ is the disturbance feedforward control transfer function. Theoretically, $G_r$ may be designed as:

$$G_r = G^{-1}$$  \hspace{1cm} (3.3)

to take care of changes in reference values to achieve perfect tracking, and $G_d$ as:

$$G_d = -G^{-1}G_t$$  \hspace{1cm} (3.4)

to take care the interaction effects.

Feedforward and feedback control complement each other. Feedforward actions are meant to perform fast corrections due to changes in the reference value or in the disturbance. Feedback gives corrective action on a slower time scale to compensate for inaccuracies in the process model, measurement errors, and unmeasured disturbances.

Feedforward can compensate for some of the limitations of feedback, such as:

- Corrections are made until after a deviation occurs in the measured variable; it is theoretically impossible to achieve perfect control during load and set-point changes.
- Cannot compensate for known disturbances in a predictive way.
- May not be satisfactory in systems with long time constants or long time delays; under large and frequent disturbances the process may operate continually in a transient state, without never achieving true steady state.
Difficulties with feedforward include:

- The load disturbance must be measured on-line; in many applications this is not feasible.
- A model of the process is needed; the quality of the feedforward action depends on the accuracy of the process model.
- The feedforward controller often contains pure derivatives that cannot be realized in practice.

Advantages of feedforward include:

- A fast corrective action can be made in a predictive way if the disturbance can be measured.
- Approximations to ideal pure derivatives often provide effective control.

Advantages of feedback include:

- Regardless of the source and type of disturbance, corrective action occurs as soon as the controller deviates from the set-point.
- A model of the process does not need to be perfectly known; compensation can be made for model inaccuracies and dynamics not modeled.

For application in an industrial multivariable process, the single feedforward-feedback control loop can be easily extended to compensate for any number of interaction effects due to the feedback control actions and set-point changes from other control loops. Then, the transfer function (3.2) becomes:

\[
Y_k = [1 + G_k G_{ek}]^{-1} \sum_j (G_k G_{ek} + G_k G_{rjk}) R_j + \sum_{j \neq k} (G_{ijk} + G_k G_{djk}) U_j \tag{3.5}
\]

where \(k = 1, \ldots, p\) is the control loop number at hand, \(j = 1, \ldots, m\) is the control loop number, \(Y, R,\) and \(U\) are the Laplace transforms of the output, set-point, and control signals, respectively, \(G_k, G_{ek},\) and \(G_{rk}\) are the process, feedback control, and reference feedforward control transfer functions in loop \(k\), respectively, while \(G_{ijk}\) and \(G_{djk}\) are the interaction, and disturbance feedforward transfer functions from loop \(k\) to loop \(j\), respectively. All the control loops of the form (3.5) can be arranged in the multivariable feedforward-feedback control scheme of the ICCS-MP. To that aim, Figure 3.14 shows
the layout for a two-input-two-output process as would be obtained directly from Figure 3.13, where the transfer functions of the feedforward interaction disturbance elements have been grouped in the so called interaction compensator.

Through adequate definitions the system in Figure 3.14 is successively redrawn in Figure 3.15, where:

\[
G_{fb} = G_d G_c
\]  
(3.6)

\[
G_{ff} = G_d G_r
\]  
(3.7)

would be the feedback and feedforward control path transfer matrices, respectively, of an equivalent linear version of the ICCS-MP control scheme in Figure 3.10.

Further extension of the control scheme to the nonlinear and wide-range operation case for a fossil fuel power unit requires additional considerations, and it is carried out throughout the following chapters. Meanwhile, in the next section, it is shown how the proposed feedforward-feedback control scheme relates to, and extends, the structure of a current coordinated control scheme in a fossil-fuel power plant.
3.4.5 Feedforward requirements

A FFPU is a large complex system for which the plant transfer function, $G(s)$, if known, is only valid around a single operating point, and its inverse, $G^{-1}(s)$, cannot be guaranteed to exist or to be causal. Hence the ideal feedforward/feedback strategy as presented in the previous section, could be inadequate to attain wide-range operation. Nevertheless, the requirements on the feedforward control $G_{ff}(s)$ can be lessened to only approximate the plant inverse dynamics, and let the feedback control path compensate for
uncertainty when tracking any unit load load profile. However, there is still the need for a plant model, which is a particularly difficult issue for control with distributed systems. To overcome this problem it is proposed for the feedforward control path to approximate the inverse steady-state behavior of the FFPU. To this aim, the feedforward control will be implemented as a MIMO fuzzy system that solves the plant inverse kinematic problem using input-output process data along its whole operating range, that is, with no process model strictly required for its design.

3.5 Extended Coordinated Control

At fossil fuel power units, the coordinated control (CC) scheme is the uppermost layer of the control system. The CC is responsible for driving the boiler-turbine-generator set as a single entity, harmonizing the slow response of the boiler with the faster response of the turbine-generator, to achieve fast and stable unit response during load changes and load disturbances. Typically, the CC embraces the power and pressure control loops (Figure 3.16). Given a unit load demand, $E_{uld}$, the CC provides demands to the control

![Figure 3.16 Conventional coordinated control.](image-url)
loops. Ordinarily, the set-point for the power control loop, $E_{d_t}$, is equal to the unit load demand, and the set-point for the pressure control loop, $P_{d_t}$, is obtained through a non-linear mapping that implements a variable pressure operating policy.

Despite the variations in the implementation of actual CC schemes by different manufacturers, a typical CC scheme basically consists of a decentralized multiloop configuration of SISO PI-based feedback control loops with constant parameters. Also, practical CC implementations show complementary feedforward compensation schemes designed to enhance the system response and to minimize the effect of interaction between the control loops due to the coupled dynamics of the process [Taft 1987]. Most realizations have evolved through various decades of research and practical experience, and most of them are confidential industrial property of their developers. Current CC schemes are seriously challenged during wide-range load-following operation. Unit performance may decrease due to large nonlinear variations and coupling effects of the process dynamics, for which there is no general and systematic way to implement control loop compensations. Also, with current CC schemes it is not possible to optimally satisfy multiple conflicting operation objectives, as is currently required for power units. Interestingly, there are no mechanisms available to specify the requirements of an operating scenario, and to incorporate them into the process optimization strategy.

Primarily, the ICCS-MP is intended to extend the scope of current coordinated control schemes in fossil fuel power plants (Figure 3.17). The ICCS-MP addresses the above mentioned needs, and conforms itself to a general structure that will allow the control system to satisfy even more complex automation requirements.

The reference governor generates set-point trajectories for the lower level control loops by solving a multiobjective optimization problem, for which the objective functions and their priorities can be set arbitrarily, in number and form. This approach allows for process optimization, and provides a way to specify the operating policy to accommodate a great diversity of operating scenarios.

As previously explained, the feedforward-feedback control scheme in the ICCS-MP is an extension of the general linear single-input-single-output feedback control
scheme, with both reference feedforward and disturbance feedforward actions, to the multivariable case. The feedforward control processor is implemented as a set of MIMO fuzzy inference systems designed from plant input-output data using a neural network paradigm. This approach provides the control system with off-line learning capabilities to attain process optimization under changing operating conditions.

Figure 3.17 ICCS-MP as extended coordinated control.
The feedback control path is implemented as a PID-based decentralized (multiloop) control scheme with a disturbance feedforward interaction compensator, which is designed using the relative gain array technique. Both the control algorithms and the compensator are first order Sugeno-type fuzzy inference systems, scheduled in two dimensions (power and pressure) to achieve satisfactory disturbance rejection and uncertainty compensation during wide-range operation. The process operating window is partitioned to take into account the process nonlinear characteristics, and tuning is carried out by a genetic algorithm at the points of interest in the partitions.

3.6 Summary

This chapter presented the design of the ICCS and the ICCS-MP. The ICCS is conceived as a multiagent intelligent system that tackles the complexity of the design of a large-scale control system in a systematic and effective way. The software engineering agency concept is proposed as the fundamental design premise for implementing an intelligent hybrid overall control system for FFPUs. The ICCS organization is an open superset of functionally grouped agent clusters. Intelligence functions are used as guidelines for clustering, which could be equivalent to location of knowledge processing zones in an artificial brain-like computer.

The ICCS-MP constitutes a minimum prototype of the ICCS, intended for realization and to demonstrate the feasibility of the ICCS paradigm, taking into account the currently available hardware and software technology. The ICCS-MP realizes the ICCS main characteristics within a reasonable scope and satisfies its main requirements through a two-level hierarchical intelligent hybrid coordinated control system for FFPUs.
CHAPTER 4

REFERENCE GOVERNOR

In this chapter, the reference governor module of the ICCS-MP is described. In Section 4.1, the proposed approach is briefly justified and the reference governor is overviewed. Section 4.2 presents some remarks on multiobjective optimization, as required by the reference governor. Section 4.3 presents the non-linear goal programming method, which is used to solve the multiobjective optimization problem to design the set-point scheduler in the reference governor. In Section 4.4, the proposed method to generate optimal set-point trajectories is described in detail. In Section 4.5, the results of simulation experiments carried out to demonstrate the feasibility of the reference governor are presented. Finally, Section 4.6 summarizes this chapter.

4.1 Introduction

The current operating context of a fossil fuel power unit is characterized by many needs and requirements. Firstly, a FFPU must support the main objective of the power system, which is to meet the load demand for electric power at all times, at constant voltage and at constant frequency [Elgerd 1971]. In addition, competition among utilities and other market driven forces have increased the usage of FFPUs in load-following duties [Armor 1985]. Moreover, stringent requirements on conservation and life extension of major equipment, and regulations on reduced environmental impact have to be fulfilled [Divakaruni and Touchton 1991]. This situation may be synthesized as an essential requirement for an FFPU to achieve optimal operation under multiple operation objectives, such as minimization of load tracking error, minimization of fuel consumption and heat rate, maximization of duty life, minimization of pollutant emissions, etc. Thus,
although load-tracking, and voltage and frequency stability have been the basic issues of concern, more effective control systems must also be designed to optimally satisfy arbitrary multiple, generally conflicting, operating objectives, so that the FFPU can successfully participate in the everyday more demanding structure of a power system.

From the automation point of view, attainment of optimal process operation through automatic control considers two great avenues: supervisory steady-state optimization control and dynamic optimal feedback control. Normally, supervisory controls determine the process operating conditions to command the lower level automation functions by solving optimization problems. The hierarchical structure of power systems seems to favor supervisory optimization of power units via set-point scheduling. Unfortunately, little attention has been paid in this regard. Most research has focused to achieve better feedback control; sometimes assuming that satisfactory set-point values are available, and most times ignoring that feedback control alone cannot refine operation beyond what is established by the set-points. In general, there is a lack of questioning on the origin and adequacy of the set-points for optimal operation of FFPUs.

There are only a few strategies for power plant supervisory optimization available in the technical literature. In [Dieck-Assad, Masada, and Flake 1987], sub-optimal set-point values are calculated using the dynamic model of a power unit as a constraint for the optimization of a single objective function. In [Garduno and Lee 1997] a fuzzy inference system generates pressure set-points to minimize steam throttling losses during cyclic operation. In [Prasad, Swidenbank and Hogg 1999] the set-points are shifted according to the statistical behavior of selected output signals to improve economic performance. The use of power-pressure nonlinear relationships to schedule the pressure set-point to accommodate up to three different operating conditions is shown in [Landis and Wulfsohn 1988]. It is important to note that in all these cases, there is no established mechanism to specify the requirements of the operating scenario, and consequently it is not possible to incorporate them into the process optimization strategy. Moreover, there is no provision to satisfy multiple operation objectives simultaneously, as currently required at power units.
At fossil fuel power units, the coordinated control (CC) scheme constitutes the uppermost layer of the control system. It is responsible for driving the boiler-turbine-generator set as a single entity. Typically, the CC scheme includes the power and steam pressure control loops, which account for the dominant behavior of the power unit, therefore the CC constitutes the primary means to achieve process optimization. Given the unit load demand, $E_{uldb}$, the CC provides demands to the power and pressure control loops to harmonize the slow response of the boiler with the faster response of the turbine-generator; the goal being to achieve fast and stable unit response during load changes and load disturbances. Ordinarily, the set-point for the power control loop, $E_{db}$, is equal to the unit load demand, and the set-point for the pressure control loop, $P_{d}$, is obtained from the unit load demand through a non-linear mapping along the whole power operating-range of the unit, as was shown in Figure 3.12. This mapping defines the operating policy of the unit and stays fixed in most installations. This approach does not allow for process optimization should the operating scenario changes from that considered in the original design. Hence, in view of the current operating context for FFPUs, it is of the highest interest to have the means to adjust the power-pressure mapping to optimally accommodate different operating scenarios.

In this chapter, the reference governor (RG) module of the ICCS-MP is presented (Figure 4.1). The purpose of the RG module is to provide the means to achieve multiobjective optimal process optimization under diverse operating scenarios through set-point scheduling. The RG module builds upon the concept, and extends the scope, of current coordinated control schemes regarding set-point generation. Basically, the RG module consists of a one-to-many mapping and a supervisory mapping-designer. The mapping provides all required set-points $y_{d}$ for the control loops under consideration from the unit load demand $E_{uldb}$. The supervisory designer determines the components of the mapping in an optimal way by solving a multiobjective optimization problem in accordance to the specification of the operating scenario at hand.

To achieve an open design for the RG, the one-to-many mapping is proposed as a structure which can be expanded or contracted, as required by the scope of the current
coordinated control application. In general, the one-to-many mapping will consist of several independent one-to-one mappings, each of them from the unit load demand to a single set-point. For the ICCS-MP case, the mapping consists of three one-to-one mappings (Figure 4.2), which provide the set-points for the power, $E_d$, steam pressure, $P_d$, and drum water level deviation, $L_d$, control loops, in terms of the unit load demand $E_{uld}$.

The design of all the one-to-one mappings is carried out simultaneously by the supervisory designer using the power unit static model and the specification of the operating scenario in terms of objective functions $J=[J_1\ J_2\ \ldots\ J_n]^T$ and their associated preferences $\beta=[\beta_1\ \beta_2\ \ldots\ \beta_n]^T$ following a three step design procedure, as will be defined later. Central to the design procedure is the solution of a multiobjective optimization problem that is formulated based on the goal programming approach [Rao 1996], which allows the selection of a single design solution from the set of all multiobjective optimal solutions (Pareto set).

The proposed RG contributes a method to attain process optimization through set-point scheduling under different operating scenarios characterized by multiple competing operating requirements, as currently expected at power units. An operating scenario, verbally described by various operating objectives, can be accommodated by specifying an operating policy, which can be quantitatively described in terms of objective functions.
and their respective relative preference values. In turn, the objectives and preferences are used to optimally design the unit load demand to set-point mappings, which will schedule the set-points to optimally drive the power unit through multiobjective optimal operating points along any required unit load demand pattern. The feasibility of the proposed approach is demonstrated through a case study.

4.2 Multiobjective Optimization

Motion planning, resource allocation, activities scheduling, and system design are a few examples of engineering problems commonly seeking the best possible solution. Inherent to any solution methodology, there is an optimization problem to be solved. Most real-world optimization problems are multicriterion in nature, where the goal is to minimize or maximize not a single but several objective functions simultaneously. In this section, fundamental issues regarding multiobjective optimization are presented. The first part introduces definitions of basic terms and the nomenclature to be employed. The second part presents a general formulation of the multiobjective optimization problem in mathematical terms.
### 4.2.1 Basic terminology

The numerical quantities whose values are to be chosen during the optimization process are called the decision variables. On the other hand those quantities whose values remain fixed are simply called parameters. In a problem with \( m \) real decision variables each solution alternative is represented as a vector:

\[
x = [x_1, x_2, \ldots, x_m]^T
\]

where \( x_i \in \mathbb{R} \), for \( i = 1, 2, \ldots, m \) are the decision variables. The space of decision variables (SDV) is the \( m \)-dimensional Euclidean (real) space where the vectors of decision variables take values. In general \( \text{SDV} = \mathbb{R}^m \).

The real functions of the decision variables, used as criteria to compare the proposed solutions during the optimization process are the objective functions. In an \( n \)-dimensional multiobjective optimization problem, objective functions are also grouped in vectors:

\[
J(x) = [J_1(x), J_2(x), \ldots, J_n(x)]^T
\]

where \( J_i(x) \in \mathbb{R} \), for \( i=1, 2, \ldots, n \). The space of objective functions (SOF) is the \( n \)-dimensional Euclidean (real) space where the vectors of objective functions take values. In general \( \text{SOF} = \mathbb{R}^n \). Also, the order of components in \( x \) and \( J(x) \) is arbitrary, but once chosen it is not allowed to change, and every point \( x \) in SDV determines a point in \( J(x) \) in SOF.

The constraint functions are the relationships among the decision variables and parameters which must be satisfied in order to produce an acceptable solution. There is inequality constraints:

\[
g_i(x) \leq 0, \quad i = 1, 2, \ldots, p
\]

and equality constraints:

\[
h_i(x) = 0, \quad i = 1, 2, \ldots, q
\]
which may also be expressed as vectors to compact notation:

\[ g(x) = \begin{bmatrix} g_1(x) & g_2(x) & \cdots & g_p(x) \end{bmatrix}^T \leq 0 \tag{4.5} \]

\[ h(x) = \begin{bmatrix} h_1(x) & h_2(x) & \cdots & h_q(x) \end{bmatrix}^T = 0 \tag{4.6} \]

where \( g_i(x) \in \mathbb{R} \), for \( i = 1, 2, \ldots, p \) are the inequality constraint functions, \( h_i(x) \in \mathbb{R} \), for \( i = 1, 2, \ldots, q \) are the equality constraint functions, and \( g(x) \geq 0 \) and \( h(x) = 0 \) denote array (element by element) relationships in \( \mathbb{R}^p \) and \( \mathbb{R}^q \), respectively.

The feasible region in the decision space (FRDS) is the set of all decision vectors in SDV that satisfy the equality and inequality constraints:

\[ \text{FRDS} = \{ x \in \text{SDV} : g(x) \geq 0 \text{ and } h(x) = 0 \} \tag{4.7} \]

where every point \( x \in \text{FRDS} \) is a feasible solution alternative.

The feasible region in the objective space (FROS) is the image of FRDS in the space of objective functions SOF:

\[ \text{FROS} = \{ y \in \text{SOF} : y = J(x) \text{ and } x \in \text{FRDS} \} \tag{4.8} \]

where every point \( x \in \text{FRDS} \) is a feasible solution alternative.

The set FRDS is convex if for every two points in FRDS, the line segment joining them lies completely in FRDS. That is, if for every \( x_1, x_2 \in \text{FRDS} \), and every \( \lambda \in [0,1] \), it holds:

\[ \lambda x_1 + (1-\lambda)x_2 \in \text{FRDS} \tag{4.9} \]

A real-valued objective function \( J \) on the convex set FRDS is said to be convex on FRDS if for every \( x_1, x_2 \in \text{FRDS} \), and every \( \lambda \in [0,1] \), it holds:

\[ J(\lambda x_1 + (1-\lambda)x_2) \leq \lambda J(x_1) + (1-\lambda)J(x_2) \tag{4.10} \]

A multiobjective optimization problem is said to be convex if and only if the following conditions are satisfied:

- The components of the vector of objective functions \( J(x) \) are convex.
- The components of the vector of inequality constraints \( g(x) \) are convex.
The components of the vector of equality constraints \( h(x) \) are affine-linear functions of \( x \).

Convexity attributes could be hard to determine due to the non-linearity of the objective functions and/or constraints. To prevent this situation, this dissertation proposes to split any complex objective function into several simple convex objective functions, and constraints are inequalities in the form of convex open sets, as will be shown later.

### 4.2.2 General formulation of the multiobjective optimization problem

In its most general formulation, the multiobjective optimization problem (MOOP) is usually stated in the form [Sawaragi, et al. 1985, Miettinen 1999]:

**Minimize:**

\[
\{ J_1(x), J_2(x), \ldots, J_n(x) \} 
\]

subject to:

\[
x \in SDV
\]

\[
g(x) \leq 0
\]

\[
h(x) = 0
\]

where: \( x \) is an \( m \)-dimensional vector of decision variables, \( J_i(x) : \mathbb{R}^m \rightarrow \mathbb{R} \) for \( i = 1, 2, \ldots, n \), are the objective functions, \( g(x) \) is a \( p \)-dimensional vector of inequality constraint functions, and \( h(x) \) is a \( q \)-dimensional vector of equality constraint functions. \( J(x) = [J_1(x) J_2(x) \ldots J_n(x)]^T \) is an \( n \)-dimensional vector of objective functions defined in SOF.

The term “minimize” used in (4.11a) indicates that all the objective functions are to be minimized simultaneously. In general, for each objective function a different solution is obtained. An optimal solution that minimizes all the objective functions simultaneously does not necessarily exist; thus the objective functions are said to be conflicting, and incommensurable if expressed in different units. This situation makes it necessary to introduce the concept of non-dominated solutions or non-inferior solutions of a MOOP, also known as Pareto optimal solutions.
Intuitively, a feasible point is said to be Pareto optimal if when perturbed to improve (reduce) one objective at least another criterion gets worse (increases). Formally, a point \( x^* \in \text{FRDS} \) is said to be Pareto optimal if and only if for some neighborhood of \( x^* \) there is no point \( x = x^* + \Delta x \) such that the following conditions are satisfied:

\[
\begin{align*}
    x & \in \text{FRDS}, \\
    J_i(x) & < J_i(x^*) \quad \text{for at least one } i \in \{1, 2, \ldots, n\}, \text{ and} \\
    J_i(x) & \leq J_i(x^*) \quad \text{for all } i \in \{1, 2, \ldots, n\}
\end{align*}
\] (4.12a) (4.12b) (4.12c)

In general, a MOOP does not have a unique Pareto-optimal point. The set of all points that are Pareto-optimal for a given MOOP is called the Pareto-optimal set (POS):

\[
\text{POS} = \{x^* \in \text{FRDS} : x^* \text{ is Pareto optimal}\} 
\] (4.13)

Correspondingly, the set of all points in the space of objective functions that are images of the Pareto-optimal points is called the optimal contour set (OCS):

\[
\text{OCS} = \{y \in \text{FROS} : y = J(x^*)\} 
\] (4.14)

### 4.2.3 MOOP solution methods

From a mathematical point of view, the MOOP is considered to be solved once the Pareto optimal set is found or a great portion of it is known so that more solutions can be obtained by inspection or simple interpolation between known solutions. Nevertheless, practical applications require additional considerations. First, it is usually required to select a single solution from the set of all Pareto optimal solutions, for which an extra decision making process must be carried out. Second, although the Pareto optimal set determines the optimal contour set through \( J(x) \), in practice the POS is determined from the OCS, which can be first found by many different methods. Furthermore, also due to practical convenience, the most general desired requirements for a solution strategy to satisfy are: 1) to allow expression of the MOOP in a numerically tractable way, perhaps different from the general formulation provided in (4.11), and 2) to provide the means to
define preferences among the objectives being considered. It is highly desirable that both requirements be met in an intuitive and natural way, that is, the formulation should allow a meaningful expression of the optimization procedure requirements, and the preferences should effectively reflect priorities among the optimization objectives without ambiguities.

Among the most well-known methods currently available to solve a MOOP [Rao 1996] are: the utility function, the inverted utility function, the global criterion, the bounded objective function, the lexicographic, and the goal programming methods. The utility function method is very attractive for practical applications due to the simplicity and intuitiveness of its formulation, and because it can be solved using a standard scalar optimization algorithm. This method reformulates the MOOP statement in (4.11) into a single objective optimization problem by defining a total or overall utility function, $U(J)$, in terms of the individual utility functions, $U_i(J_i)$, defined for each objective. Then, the solution is found by maximizing or minimizing, as required, the overall utility function subject to the constraints. The weighted sum strategy is one of the most appealing and simplest forms of the utility function method. In this case, the MOOP is formulated as:

Find $x$ that minimizes:  

$$ U = \sum_{i=1}^{n} w_i J_i^2(x) $$  

(4.15a)

such that:  

$$ x \in SDV, \ g(x) \geq 0, \ and \ h(x) = 0 $$  

(4.15b)

where the $w_i$ are scalar weighting factors whose values are set to assign preferences to the associated objective functions $J_i(x)$.

Despite its attractiveness, the weighted sum strategy has two major drawbacks. First, it is not a trivial problem to assign the values of the weighting factors since the coefficients do not necessarily correlate with the relative importance of the objectives, thus making it difficult to express trade-offs between the objectives based on their preferences. This issue gets unnecessarily more complicated when the objectives use different engineering units and also when the number of objectives gets large. Second, the weighted sum strategy does not guarantee access to all the multiobjective optimal solutions. This happens when the optimal contour set (OCS) presents a non-convexity
facing the origin of the space of objective functions (SOF). To overcome these negative aspects, while preserving the simplicity and intuitiveness of the formulation, as well as the convenience of the numerical manipulation, a non-linear goal programming approach is used in this work.

4.3 Goal Programming Method

As with the utility function method, the goal programming (GP) method also reformulates the MOOP in (4.11) to end up with a single objective optimization problem. The basic strategy of the GP method is to look for a solution $x$ to produce an objective vector, $J(x)$, as close as possible to a target objective vector, $J_t$. To formulate this strategy in a numerically tractable form, what is minimized is the distance, $d(J(x), J_t)$, from the candidate solution $J(x)$ to the target vector $J_t$, which is expressed as a function of the deviation $\delta = J_t - J(x)$, and is then called an achievement function, $h(\delta)$, by the different variations of the GP method.

4.3.1 Goal programming basics

To avoid ambiguity in its formulations, the GP approach introduces some specific terminology. First, an objective is a mathematical function of the decision variables that represents some desire or wish of the decision maker. The two most common objective function forms are:

$$\text{maximize } f(x) \text{ or minimize } f(x)$$

where the value of the objective function is left unspecified. Unless otherwise specified only the minimization form of objectives will be used hereafter. Complementarily, an aspiration level or target level is a specific (realistic) value associated with a desired or acceptable level of achievement of an objective. Thus, an aspiration level may be used to
measure the achievement or non-achievement of an objective. The pair of an objective and an aspiration level is called a goal. The general form of a goal, for an objective in the minimization form, is:

\[ \text{satisfy } f(x) \leq b \] (4.17)

where \( b \) is the aspiration level used to measure the achievement of the goal. A constraint is an inflexible (also called rigid, absolute, or hard) goal, and has the same mathematical appearance as a goal. In goal programming, the concept of a constraint is considered a subset of the concept of goals, in this way, the GP method works solely with goals; some of them flexible (goals from objectives), and some rigid (goals from constraints).

The general GP optimization model converts the basic MOOP in (4.11) to an alternate model consisting only of goals by establishing aspiration levels, \( J_{ti} \), for all the objectives:

\[ \text{minimize } J_i(x) \Rightarrow \text{satisfy } J_i(x) \leq J_{ti} \text{ for } i=1, 2, \ldots, n \] (4.18)

where additional conditions are set for by the specific GP method formulations, which differentiate among themselves by the way they measure the attainment of the set of goals in (4.18).

As mentioned in the beginning of this section, goal attainment, or goodness of a candidate solution, is measured in terms of the goal deviations:

\[ \delta_i = J_{ti} - J_i(x) \text{ for } i=1, 2, \ldots, n \] (4.19)

where the goal deviations form a goal deviation vector \( \delta = [\delta_1 \delta_2 \ldots \delta_n]^T \), and the aspiration levels \( J_{ti} \) form an aspiration level vector that corresponds to the target objective vector \( J = [J_{t1} J_{t2} \ldots J_{tn}]^T \). Also, as a distinctive characteristic of the GP method, the goal attainment measurement is carried out through a scalar achievement function of the goal deviation, \( h(\delta) \), which transforms the vectorized MOOP into a single objective optimization problem, and may be minimized using any appropriate scalar optimization algorithm available. To this aim, it is desired for the achievement function to be a monotonically increasing function of the goal deviation. As will be shortly shown, this
requirement can be easily satisfied if each component of the goal deviation vector, which may be either positive or negative, is expressed as the difference of two other positive-valued deviations:

\[ \delta_i = \delta_{ni} - \delta_{pi} \quad \text{for } i = 1, 2, \ldots, n \] (4.20)

where \( \delta_{pi} \geq 0 \) and \( \delta_{ni} \geq 0 \) are respectively called the positive and negative deviation terms of the \( i \)-th objective. It is important to note that given a non-zero deviation, only one of the two deviation terms is non-zero, that is \( \delta_{ni}\delta_{pi}=0 \) always holds, therefore combining (4.19) and (4.20), it can be seen from:

\[ J_i(x) + \delta_{ni} - \delta_{pi} = J_{ti} \quad \text{for } i = 1, 2, \ldots, n \] (4.21)

that \( \delta_{ni} \) measures the underachievement, \( J_i(x)<J_{ti} \), and \( \delta_{pi} \) the overachievement, \( J_i(x)>J_{ti} \), of a goal for any given candidate solution.

### 4.3.2 Minsum goal programming formulation

Generally, the definition of the achievement function, \( h(\delta) \), yields different methods of the goal programming approach. Defining the achievement function as a monotonically increasing \( p \)-norm in terms of the deviation terms \( \delta_{ni} \) and \( \delta_{pi} \) yields what is known as the minsum GP formulation of the MOOP:

Find \( x \) which minimizes the achievement function:

\[ h(\delta_p, \delta_n) = \left[ \sum_{i=1}^{k} (w_p \delta_{pi} + w_n \delta_{ni})^r \right]^{1/r}, \quad r \geq 1 \] (4.22a)

subject to:

\[ x \in SDV \] (4.22b)
\[ g_i(x) \leq 0 \quad \text{for } i = 1, 2, \ldots, p \] (4.22c)
\[ J_i(x) + \delta_{ni} - \delta_{pi} = J_{ti} \quad \text{for } i = 1, 2, \ldots, n \] (4.22d)
\[ \delta_{pi} \geq 0, \delta_{ni} \geq 0 \quad i=1, 2, \ldots, n \quad (4.22e) \]

where \( w_{pi} \) and \( w_{ni} \) are respectively called the positive and negative weighting factors of the \( i \)-th objective.

In general, the goal programming method minimizes a metric of the deviations from the target objectives, instead of directly minimizing the objective functions of the general formulation in (4.11) or a composed function of the objectives as in the weighted sum method in (4.15). As the utility function method, the goal programming approach is in the form of a single-objective problem, constrained by all the objective functions stated as goals, that can be numerically solved with any appropriate scalar optimization algorithm. The solution procedure will inherently include the decision making process to a unique solution out of the Pareto optimal set. The minsum GP approach is one of the most useful formulations to solve a MOOP. After a series of simplifications based on the specific characteristics of the optimization process at a FFPU, this method will be used to implement the set-point scheduler in the ICCS-MP.

### 4.3.3 Optimization algorithm for FFPU

Central to the design of the set-point mappings in the reference governor module of the ICCS-MP, will be the formulation of a MOOP in the spirit of the minsum goal programming formulation in (4.22). To this aim, some problem specific features are intuitively and successively taken into account to have an even simpler formulation and a working algorithm. First, the achievement function can be simplified setting \( r=1 \), which will initially yield an \( l_1 \)-norm. Second, since all objective functions are to be minimized, only the measures of overachievement will be meaningful; hence only the positive deviation terms and their weighting factors will be utilized, that is the summation argument will have only one term. Third, from all deviations being considered up to this point, only the worst (maximum) positive deviation term (overachievement) is necessary to be minimized. Application of these measures to (4.22) yields the formulation:
Find $x$ that minimizes:

$$\delta_m = \max_{i=1,\ldots,k} \delta_{pi}$$  \hspace{1cm} (4.23a)

subject to:

$$(J(x) - J_t) - w \delta_m \leq 0$$  \hspace{1cm} (4.23b)

where $J_t$ is a $k$-dimensional vector of target objectives, $w$ is a $k$-dimensional vector of non-negative weights, $w_i \geq 0$, to be defined below, and (4.23b) is an element-by-element inequality. Note that the resulting $l_\infty$–norm in (4.23a) might be obtained directly from (4.22a) by setting $r = \infty$, but lacking the intuitive preamble just presented.

Generally, the elements, $J_{ti}$, of the objective function target vector, $J_t$, in (4.23b) are obtained by previously solving the single-objective optimization problems:

$$J_{ti} = \min \{J_i(x) : x \in FRDS\} \quad i = 1, 2, \ldots, k$$  \hspace{1cm} (4.24)

Also, the elements, $w_i$, of the weighting coefficient vector, $w$, in (4.23b) may be chosen arbitrarily to reflect preference on the objectives. To ease this task and make it intuitive, it is proposed here to set them using:

$$w_i = (1 - \beta_i) J_{ti}$$  \hspace{1cm} (4.25)

where each $\beta_i \in [0,1]$, $i = 1, 2, \ldots, k$, is introduced as a normalized unit-less value that specifies the relative preference of the $J_i$ objective. Intuitively, setting $\beta_i = 0$ assigns the lowest relative preference to the associated objective function, whereas $\beta_i = 1$ assigns it the highest relative preference. Note that setting $\beta_i = 1$ makes $w_i = 0$ and causes the associated constraint in (4.23b) to be a hard constraint, that is $J_i(x) = J_{ti}$, must be satisfied since by definition $J_{ti}$ is the minimum value of $J_i(x)$. On the other hand, setting $0 \leq \beta_i < 1$ may be used to assign a degree of slackness in the achievement of the corresponding objective. Equal relative preference values can be assigned to two or more objective functions to indicate that they have the same preference and the same degree of slackness is allowed in their achievement. The relevance of this approach is that it facilitates dealing with incommensurable objectives, and true preference can be assigned for all objectives in a common basis.
4.4 Supervisory Mapping Designer

The essence of the problem at the reference governor in the ICCS-MP is that of designing the optimal mappings from the unit load demand $E_{uld}$ to the set-points $E_d$, $P_d$, and $L_d$:

\begin{align*}
SP_E &: E_{uld} \rightarrow E_d \\
SP_P &: E_{uld} \rightarrow P_d \\
SP_L &: E_{uld} \rightarrow L_d
\end{align*}

(4.26a, b, c)

that will be used to transform any unit load demand pattern $(E_{uld}, t)$ into optimal set-point trajectories for the power $(E_d, t)$, pressure $(P_d, t)$ and level $(L_d, t)$ control loops:

\begin{align*}
(E_{uld}, t) &\rightarrow (E_d, t) \\
(E_{uld}, t) &\rightarrow (P_d, t) \\
(E_{uld}, t) &\rightarrow (L_d, t)
\end{align*}

(4.27a, b, c)

where $t$ is time (sec).

The set-point mappings $SP$ are basically designed by solving a multiobjective optimization problem that takes into account the specified operation objectives, their relative preferences, and the steady-state model of the plant. Then, the supervisory mapping-designer performs the design process off-line in three steps (Figure 4.3):

- Determination of the feasibility regions for the decision variables.
- Solution of the multiobjective optimization problem to find optimal steady-state control signals.
- Calculation of the set-points through direct evaluation of the steady-state model of the unit.

This procedure is applied at several power values spanning the entire power operating range. Finally, the set-point mappings in (4.26a,b,c) are composed with the resulting input $(E_{uld})$ and output $(E_d, P_d, L_d)$ data. The resulting set-point mappings are programmed as look-up tables that provide data at intermediate values by interpolation.
Without any loss of generality and to ease the presentation of the set-point mapping design procedure, a case study where the objective functions depend only on the values of the control signals $u=[u_1 \ u_2 \ u_3]^T$ is presented. Extension to objective functions containing state variables, or any other system signal, follows a similar approach. The case study introduced in this section will be used again in Chapter 7 to demonstrate the feasibility of the whole ICCS-MP.

### 4.4.1 Feasibility regions of control signals

The feasibility regions, $\Omega_i, i=1, 2, 3$, for the decision variables $u_1$, $u_2$, and $u_3$, may be determined experimentally, or set manually to impose arbitrary operating constraints. In this dissertation, the nonlinear mathematical model of the FFPU was used to obtain the operating windows of the control valve demands. The process is similar to that followed to determine the power-pressure operating window of the power unit described in Section 2.4.4. The feasibility regions for the control signals $u_1$, $u_2$ and $u_3$ are shown in Figures 4.4, 4.5, and 4.6, respectively. Note that at any given power operating point the fuel and steam valve demands may vary substantially, while the feedwater valve demand shows a relatively small variation. Later, these facts will be of great relevance for optimization.

Once the feasible regions are determined, the envelopes are programmed as look-up tables that provide the upper and lower limits of the feasible regions as functions of the unit load demand value:
Figure 4.4 Fuel valve demand operating window.

Figure 4.5 Steam valve demand operating window.

Figure 4.6 Feedwater valve demand operating window.
It should be noted that for any given unit load demand, the feasible regions of the decision variables are open convex sets.

4.4.2 Optimal steady-state control signals

In the second stage, a MOOP formulated as in (4.23) is solved for a prescribed value of the unit load demand $E_{uld}$. The purpose is to find, in a multiobjective sense, an optimal vector of inputs $u=[u_1 \ u_2 \ u_3]^T$ in the feasible regions $\Omega_i$, $i=1, 2, 3$, previously determined, that minimizes the desired objective functions, $J(u)=[J_1(u) \ J_2(u) \ldots \ J_k(u)]^T$, taking into account their predefined relative preferences $\beta=[\beta_1 \ \beta_2 \ldots \ \beta_k]^T$.

As will be shortly shown, the objective functions may account for load-tracking error, thermal stress, heat rate, pollution, or any other operating objective of interest to be optimized. Furthermore, with the proposed multiobjective approach any operating objective may be accounted for with more than one objective function. This fact allows the designer to use several simple convex objective functions to account for different aspects of a single complex operating objective, for which a single convex objective function could be very difficult to formulate. Consequently, this advantageous property facilitates the formulation of a convex MOOP and increases the versatility and openness of the approach.

At the end of the second stage, the calculated vector of optimal inputs, $u^*$, and the corresponding unit load demand value being considered, determine an optimal operating point, in the specified multiobjective sense, for the plant. Solving the MOOP at several points along the power operating range will define a multiobjective optimal relationship between the unit load demand and the plant inputs:

$$E_{uld} \rightarrow u^*=[u_1^* \ u_2^* \ u_3^*]^T$$  \hspace{1cm} (4.29)
4.4.3 Calculation of set-points

In the third stage, the vector of optimal control signals, \( u^* \), is used to generate a vector of optimal set-points through the steady-state model of the power unit:

\[
[E_d \ P_d \ L_d]^T = M_{ss} \left( [u_1^* \ u_2^* \ u_3^*]^T \right)
\]  

(4.30)

where \( M_{ss} \) is the power unit steady-state model solved with \( u \) as input and the controlled variables as outputs. The steady-state model was obtained in Section 2.4 by equating the unit’s dynamic state equations (2.28, 2.29, and 2.30) to zero. Thus, using the direct static model in Section 2.4.1, the power, pressure and level set-points are given by:

\[
E_d = \frac{0.73u_1^*-0.16}{0.0018u_2^*} \left( 0.9u_1^* - 0.15u_3^* \right)
\]  

(4.31a)

\[
P_d = \frac{141u_3^*}{1.1u_3^*-0.19}
\]  

(4.31b)

\[
L_d = 6.5365\rho_f + 3000\alpha_s + 5.5556\omega_e - 3275
\]  

(4.31c)

where \( \omega_e \), \( \alpha_s \), and \( \rho_f \) are calculated with:

\[
w_e = 45.59166u_1^* + (0.8537u_2^* - 0.14746)P_d - 2.51431u_3^* - 2.0958
\]  

(4.32a)

\[
0 = 0.13073\rho_f^2 + \left( \frac{w_e}{9} - 65.5 - \frac{0.00154\rho_s}{1 - 0.00154\rho_s} \right)\rho_f + \frac{60\rho_s}{1 - 0.00154\rho_s}
\]  

(4.32b)

\[
\alpha_s = (1/\rho_f - 0.00154) / (1/\rho_s - 0.00154)
\]  

(4.32c)

and \( \rho_s \) is given by:

\[
\rho_s = 0.8P_d - 25.6
\]  

(4.32d)

Finally, as in the second stage, calculation of the controlled variables at any given unit load demand value defines a multiobjective optimal operating point for the plant. Calculation at several points along the power operating range defines the multiobjective optimal relationships that constitute the desired set-point mappings in (4.26). These mappings are programmed as look-up tables in the ICCS-MP.
4.4.4 Unit responsiveness

Unit responsiveness refers to the capacity of the power unit to undertake sudden changes in power generation based on the amount of steam energy stored in the boiler that can be released for that purpose. Thus, at any given power value, operation at higher drum pressure is said to be more responsive because there is more steam available for a rapid release, than at lower pressure. The power-pressure characteristic being used by the control system specifies the unit responsiveness.

The previous three-step design procedure can be started from any initial condition, but in doing so, the ability to establish a desired unit responsiveness can be lost, since the outcome of the optimization procedure will depend entirely on the subsequent evolution of the optimization algorithm. To avoid this situation, the initial conditions can be assigned values corresponding to an expected power-pressure mapping with the desired unit responsiveness. The constant and variable pressure characteristics shown in Figure 4.4 are reasonable choices. This approach makes the design procedure to be more that of a refinement process to get the optimal mappings with the power-pressure relationship around the desired responsiveness requirement.

4.5 Multiobjective Optimal Set-Point Mappings

In this section, the proposed design procedure is applied to the design of the set-point scheduler mappings for the ICCS-MP. The design is presented as a case study for an operating scenario where improved load-tracking and heat-rate are the major operating objectives for process optimization. First, it is shown how to accommodate the operating scenario through the specification of the optimization goals, to carry out the design procedure at a single operating point. Then, the proposed design procedure is used to construct the set-point mappings spanning the whole operating range of the power unit. Finally, the mappings are used to investigate the ideal process optimization of the power
unit during wide-range cyclic operation. To show the effect of multiple objectives being considered, the application of the design procedure is carried out in three stages or cases, each one considering more objectives than its predecessor.

4.5.1 Specification of optimization goals

Any operating scenario is first verbally defined by stating the desired operating objectives; in this case they are 1) improved load-tracking, and 2) improved heat-rate. As previously stated, the operating policy corresponding to this scenario will be the set of optimally designed mappings to schedule the set-points. The operating policy will be determined in terms of the objective functions and their relative preference values, which also define the operating scenario in a quantitative way; adequate for mathematical treatment and numerical manipulation. From the process engineering concepts in Chapter 2, it is known that to improve load tracking and heat rate the load tracking error, fuel usage, and throttling losses in the main steam and feedwater valves, should be primarily taken into account. For this purpose, the following objective functions can be considered for minimization:

\[ J_1(u) = |E_{uld} - E_{ss}| \]  
\[ J_2(u) = u_1 \]  
\[ J_3(u) = -u_2 \]  
\[ J_4(u) = -u_3 \]

where \( E_{uld} \) is the unit load demand (MW), and \( E_{ss} \) is the corresponding generation (MW) as provided by the steady-state model:

\[ E_{ss} = \frac{0.73u_2 - 0.16}{0.0018u_2} (0.9u_1 - 0.15u_3) \]  

Regarding the objective functions, \( J_1(u) \) accounts for the power generation error; its minimization will improve load-tracking, as required. The objective function \( J_2(u) \)
directly accounts for fuel consumption through the fuel valve position; minimizing \( u_1 \) will certainly reduce fuel usage. The objective function \( J_3(u) \) accounts for exergy losses due to pressure drop across the steam valve. Since the pressure drop, and consequently the exergy loss, increases as the valve closes, it is desired to keep it as wide open as possible; thus maximizing \( u_2 \), or equivalently, minimizing \(-u_2\), will reduce losses in the steam valve. A similar reasoning applies to the objective function \( J_4(u) \) which accounts for exergy losses due to pressure drop in the feedwater control valve. Minimization of the objective functions \( J_2(u) \), \( J_3(u) \), and \( J_4(u) \), as a set, account for heat-rate improvement in the power unit, as required. The proposed objective functions clearly illustrate the advantages of the approach, in that an operating requirement can be represented by one objective function, as for load-tracking, or by several simple objective functions, as for heat-rate. In general, it should be noted that more complex objective functions can always be used, perhaps to account for more specific requirements.

Next step is to assign the relative preference values to the objective functions. To this aim, and to also illustrate the effect of multiple objectives to be considered in the next section, the final operating policy will be built in three stages. In the first stage, only the minimization of the load tracking error is considered, thus only the objective function \( J_1(u) \) is to be optimized. Normally, the highest priority is assigned to this operating requirement at power plants. To reflect this fact the corresponding relative preference value must be set to the maximum, that is \( \beta_1=1 \). In the second stage, fuel usage is considered in addition to load tracking, thus both \( J_1(u) \) and \( J_2(u) \) are taken into account. The preference on load tracking is kept the same, \( \beta_1=1 \), and the relative preference for fuel usage is chosen as \( \beta_2=0.5 \) to reflect the fact that fuel consumption is less important than matching the power demand, but recognizing that minimizing fuel consumption is still relevant. Finally, in the third stage, exergy losses due to pressure drop in the steam and feedwater control valves are considered in addition to load tracking and fuel usage, that is, all four objective functions are considered. In this stage, the objectives \( J_1(u) \) and \( J_2(u) \) keep their already defined preference values. The relative preference values for \( J_3(u) \) and \( J_4(u) \) are set to \( \beta_3=1 \) and \( \beta_4=0 \), respectively. The value \( \beta_3=1 \) sets the relevance
of minimizing exergy losses in the steam valve at the same level as the minimization of
the load tracking error, which is in agreement with the importance of handling exergy
losses to improve the overall efficiency of the whole thermodynamic cycle as explained
in Chapter 2. This is also in agreement with the fact that throttling losses at the steam
control valve can be large due to its wide operating range, as was shown in Figure 4.6.
The value $\beta_4=0$ sets the relevance of minimizing the feedwater valve exergy losses at the
lowest possible value; indicating that any amount of exergy losses can be tolerated. This
condition is acceptable since exergy losses at the feedwater valve are usually negligible
in determining the overall efficiency of the power unit, as was also explained in Chapter
2. This is also in agreement with the rather narrow operating window for $u_3$, as was
shown in Figure 4.7, which indicates that exergy losses cannot be reduced substantially.

Once the objective functions and their relative preference values have been
specified, the vector of target objective values needs to be specified. As mentioned in
Section 4.3.3, the target objective values can be set by solving the single objective
optimization problems defined by each objective considered independently. In this case,
the target values can be easily established by inspection. The lowest possible value for
$J_1(u)$ is zero, which corresponds to the case of perfect load-tracking or zero load-tracking
error, hence $J_{t1}=0$. The lowest possible values for $J_2(u)$, $J_3(u)$, and $J_4(u)$ depend on
the operating point being considered, and can be directly obtained from the operating
windows of the control valves. The target value $J_{t2}$ is given by the lower limit curve in
the operating window for the fuel valve, as shown in Figure 4.5, at the desired power
operating point. The target values $J_{t3}$ and $J_{t4}$ are given by the upper limit curve in the
operating window for the steam and feedwater control valves, as shown in Figures 4.6
and 4.7 respectively, at the desired power operating point. Table 4.1 summarizes all the
parameters required to define the optimization goals in the formulation of the MOOP to
calculate the set-points at any given power operating point. Note that relations (4.28a)
and (4.28b) have been used to specify the target values in terms of the given unit load
demand, $E_{udd}$.
Once all the optimization goals have been set, the multiobjective optimal mappings are built using the procedure described in Section 4.4 with the optimization algorithm in (4.23). A vector of set-points \( [E_d, P_d, L_d]^T \) is calculated at every specified power operating point of a set spanning the whole operating range of the unit load demand \( E_{uld} \). For all design cases, a set of 18 operating points ranging from 10 MW through 180 MW spaced 10 MW apart, is used. The optimization is carried out for initial conditions in the sliding pressure operating policy defined in Table 2.7. The mappings are built for each one of the different cases considering one, two, and four objective functions, with the relative preference values as summarized in Table 4.1. The resulting mappings are shown in Figures 4.7 through 4.9.

For all cases considered, it was obtained \( E_d = E_{uld} \) and \( L_d = 0 \). These results were expected since the load-tracking error is zero for perfect tracking, and, also in steady-state, there should be no deviation in the drum water level due to plant design based on mass and energy transformations in equilibrium conditions. Regarding the pressure set-point \( P_d \), a comparison of results shows a trend to lower the pressure set-point values along the whole unit load demand range as more objectives are considered. Interestingly, the first decrease in pressure, form the 1-objective to the 2-objectives case, was obtained

<table>
<thead>
<tr>
<th>Objective</th>
<th>( \beta ) (1-obj)</th>
<th>( \beta ) (2-obj)</th>
<th>( \beta ) (4-obj)</th>
<th>( J_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J_1(u) )</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>( J_2(u) )</td>
<td>-</td>
<td>0.5</td>
<td>0.5</td>
<td>( f_{lower}(E_{uld}) )</td>
</tr>
<tr>
<td>( J_3(u) )</td>
<td>-</td>
<td>-</td>
<td>1.0</td>
<td>( f_{2upper}(E_{uld}) )</td>
</tr>
<tr>
<td>( J_4(u) )</td>
<td>-</td>
<td>-</td>
<td>0.0</td>
<td>( f_{3upper}(E_{uld}) )</td>
</tr>
</tbody>
</table>

### 4.5.2 Multiobjective optimal mappings

Once all the optimization goals have been set, the multiobjective optimal mappings are built using the procedure described in Section 4.4 with the optimization algorithm in (4.23). A vector of set-points \( [E_d, P_d, L_d]^T \) is calculated at every specified power operating point of a set spanning the whole operating range of the unit load demand \( E_{uld} \). For all design cases, a set of 18 operating points ranging from 10 MW through 180 MW spaced 10 MW apart, is used. The optimization is carried out for initial conditions in the sliding pressure operating policy defined in Table 2.7. The mappings are built for each one of the different cases considering one, two, and four objective functions, with the relative preference values as summarized in Table 4.1. The resulting mappings are shown in Figures 4.7 through 4.9.

For all cases considered, it was obtained \( E_d = E_{uld} \) and \( L_d = 0 \). These results were expected since the load-tracking error is zero for perfect tracking, and, also in steady-state, there should be no deviation in the drum water level due to plant design based on mass and energy transformations in equilibrium conditions. Regarding the pressure set-point \( P_d \), a comparison of results shows a trend to lower the pressure set-point values along the whole unit load demand range as more objectives are considered. Interestingly, the first decrease in pressure, form the 1-objective to the 2-objectives case, was obtained.
without the intervention of $J_3(u)=-u_2$ which directly accounts for losses in the steam valve; further downward shifting was obtained by considering $J_3(u)$ explicitly in the 4-objective case. This behavior confirms qualitatively that process optimization can be achieved, in general, by opening the steam throttling valve as wide as possible for the given operating conditions. While operators do this intuitively in actual plants without any guarantee, this method provides a specific value for the pressure set-point such that all imposed operating constraints are optimally satisfied.

Also, the multiobjective-optimal control signals in steady-state conditions are presented in Figures 4.10 through 4.12. Regarding the demand to the fuel control valve, $u_1$, it should be noted that its best (lowest) values were obtained in the 2 objective case, when $J_2$ is considered. No further improvement in load-tracking and fuel usage is obtained with the aggregation of $J_3$ and $J_4$. Nevertheless, the case with 4-objectives shows that overall process optimization can still be improved by combining the demands to the steam and feedwater control valves.

### 4.5.3 Performance limit of process optimization

The results of simulation experiments presented in this section are intended to preview the operation of the power unit to achieve process optimization during wide-range cyclic operation, using the mappings just obtained. These results are ideal since they are obtained under the assumption that no process dynamics are involved, that is using only the power unit static model. The relevance of this results is that they constitute the theoretical high-limit on the unit’s performance. Results using the dynamic model are shown in Chapter 7.

The desired unit load demand pattern, $E_{ula}(t)$, consists of a cycle with small, medium, and large load changes at slow, medium, and fast rates, respectively (Figure 4.13). This ad-hoc pattern resembles a typical daily cycle for a FFPU carrying out load-tracking duties for frequency regulation as commanded from an energy control center.
Figure 4.7 Multiobjective-optimal $E_{uld}-E_d$ relationships.

Figure 4.8 Multiobjective-optimal $E_{uld}-P_d$ relationships.

Figure 4.9 Multiobjective-optimal $E_{uld}-L_d$ relationships.
Figure 4.10  Multiobjective optimal fuel valve control signals.

Figure 4.11  Multiobjective optimal steam valve control signals.

Figure 4.12  Multiobjective optimal feedwater valve control signals.
The corresponding power, pressure, and level set-point trajectories for the cases with 1, 2, and 4 objectives are shown in Figures 4.14, 4.15, and 4.16, respectively. Also, Figures 4.17, 4.18, and 4.19 show the corresponding control signals \( u_1 \), \( u_2 \), and \( u_3 \), which relate directly to the objective functions \( J_2(u) \), \( J_3(u) \), and \( J_4(u) \), respectively.

To have a better appreciation of these results, the simulation accumulated values for each one of the four objective functions are provided in Table 4.2, where as is usual for minimization problems, a smaller value indicates better performance, including the negative values for \( J_3(u) \) and \( J_4(u) \), for which the negative values reflect the definition of the objective functions (4.33c) and (4.33d). Objectives not subject to optimization are provided within parenthesis for each case; their values are presented so that all cases can be compared back to back. Unlikely, all objectives improved as the number of objectives increased. In general, results show agreement with the expected behavior.

**TABLE 4.2  CUMULATIVE VALUES OF OBJECTIVE FUNCTIONS**

<table>
<thead>
<tr>
<th>Optimization criteria</th>
<th>1-objective</th>
<th>2-objectives</th>
<th>4-objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>( J_1(u) )</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>( J_2(u) )</td>
<td>(18540.00)</td>
<td>17405.00</td>
<td>17405.00</td>
</tr>
<tr>
<td>( J_3(u) )</td>
<td>(-25425.27)</td>
<td>(-27662.25)</td>
<td>-32268.22</td>
</tr>
<tr>
<td>( J_4(u) )</td>
<td>(-17405.59)</td>
<td>(-14275.94)</td>
<td>-16656.64</td>
</tr>
</tbody>
</table>

Figure 4.13  Unit load demand cyclic pattern.
Figure 4.14  Power set-point $E_d(t)$ trajectories.

Figure 4.15  Pressure set-point $P_d(t)$ trajectories.

Figure 4.16  Level set-point $L_d(t)$ trajectories.
Figure 4.17 Fuel valve demand $u_1(t)$ trajectories.

Figure 4.18 Steam valve demand $u_2(t)$ trajectories.

Figure 4.19 Feedwater valve demand $u_3(t)$ trajectories.
4.6 Summary

This chapter contributed a method to attain process optimization through set-point scheduling at FFPUs. The method provides a mechanism to specify the requirements of the operating scenario and to incorporate them into the process optimization strategy. It also makes provision to optimally satisfy multiple operation objectives simultaneously, as currently expected at power plants. An operating scenario, verbally described by various operating objectives, can be accommodated by specifying an operating policy, which can be quantitatively described in terms of objective functions and their relative preferences. In turn, the objectives and preferences are used to optimally design the unit load demand to set-point mappings, which will schedule the set-points to optimally drive the power unit along any required unit load demand pattern. A case study was used to present the method and its feasibility was demonstrated through simulation experiments.
CHAPTER 5

FEEDFORWARD CONTROL PROCESSOR

In this chapter, the feedforward control processor module of the ICCS-MP is described. In Section 5.1, the proposed approach is briefly justified and the feedforward control processor module is overviewed. Section 5.2 presents some remarks on the neuro-fuzzy paradigm, as required for the design of the feedforward control. Section 5.3 presents the adaptive-neuro-fuzzy inference system technique, which is used to design the neuro-fuzzy systems in the feedforward control. In Section 5.4, the proposed method to build the neuro-fuzzy systems is described in detail. In Section 5.5, some simulation experiments are carried out to demonstrate the feasibility of the feedforward control processor module. Finally, Section 5.6 summarizes this chapter.

5.1 Introduction

Effective participation of a Fossil Fuel Power Unit (FFPU) in wide-range load-following duties requires the ability to undertake large power variations in the form of daily, weekly, and seasonal cycles, as well as random fluctuations about those patterns [Armor 1985, Divakaruni and Touchton 1991]. Currently, most control systems at FFPUs are multiloop configurations of PID controllers. Such approach has proved its value during normal operation at base load, where plant characteristics are almost constant, nearly linear and weakly coupled. Conversely, wide-range operation imposes strong physical demands on unit equipment, and leads inherently to conflicting operational and control situations, since FFPUs were designed to operate at constant load conditions. Under these circumstances, traditional control schemes, designed and tuned for regulation
and disturbance rejection, but for set-point tracking, may decrease the global performance of the unit; making them less acceptable for wide-range cyclic operation.

Feedforward and feedback (FF/FB) control may be combined to attain wide-range operation. The basic idea is to use open-loop feedforward control to achieve set-point driven wide-range maneuverability and closed-loop feedback control to regulate and to overcome uncertainties and disturbances around the commanded trajectory. A FF/FB control strategy has been successfully applied for wide-range load control of steam and gas turbine units in combined cycle power plants [Uram 1977, Garduno and Sanchez 1995]. In addition, some simulation-based studies have explored the FF/FB strategy using different approaches. In [Weng and Ray 1997] a robust control approach for a steam power plant is reported, in [Johns 1995] a scheme for optimal temperature tracking is presented, and a design based on genetic algorithms is presented in [Zhao, et al. 1997].

In this chapter, the feedforward control processor (FFCP) module of the ICCS-MP is presented (Figure 5.1). The purpose of the FFCP module is to facilitate wide-range set-point driven operation for the FFPU, and to provide off-line operator-requested

![Figure 5.1 Feedforward control processor in the ICCS-MP.](image-url)
system adaptability to achieve optimal operation. The FFCP module consists of a feedforward controller and a supervisory designer. The feedforward controller is an open non-linear MIMO fuzzy compensator that implements the inverse static model of the FFPU; having the set-points, $y_d$, as inputs, it provides the feedforward control signals, $u_{ff}$, as outputs. The supervisory designer, activated by the operator, uses a neuro-fuzzy paradigm to tune the feedforward controller off-line to fit a given set of optimal steady-state input-output training data patterns.

To achieve an open design for the FFCP, it is proposed for the feedforward controller a structure that can be easily and systematically expanded or contracted, as required by the scope of the coordinated control system application at hand. In general, the MIMO fuzzy compensator structure will consist of several independent MISO fuzzy sub-modules, one per each feedforward control signal to be generated. Each sub-module is fed with the set-points signals of all the control loops considered by the coordinated control, and implements a nonlinear multivariable function that provides a feedforward control signal in terms of the set-point values throughout the whole operating range of the FFPU. This approach guarantees wide-range applicability. For the ICCS-MP case, the feedforward controller consists of three MISO fuzzy systems, which provide the feedforward control signals for the fuel, $u_{1ff}$, steam, $u_{2ff}$, and feedwater, $u_{3ff}$, control valves, in terms of the power, $E_d$, pressure, $P_d$, and level, $L_d$, set-points (Figure 5.2).

![Figure 5.2 Feedforward controller in ICCS-MP.](image-url)
The fuzzy systems considered are of the Takagi-Sugeno-Kang (TSK) type [Takagi and Sugeno 1985]. The design of each fuzzy system is carried out individually, in a systematic and automated way with the Adaptive Neuro-Fuzzy Inference System (ANFIS) technique [Jang 1993] using steady-state input-output process data. If designed using the input-output data produced by the optimization procedure in the reference governor, the FFCP will be the vehicle to achieve optimal operation in a multiobjective sense, and the means to accommodate the operating scenario whenever it changes. This approach supports either off-line or concurrent design of the MISO fuzzy compensators, and provides the ICCS-MP with learning and adaptability characteristics. Numerical analysis and simulation experiments demonstrate the feasibility of the approach and show that good performance can be achieved with simple low-dimension systems, making the approach attractive for practical implementation.

5.2 Neuro-Fuzzy Paradigm

5.2.1 TSK-type fuzzy systems

In essence, a fuzzy system establishes a multi-input-single-output nonlinear mapping based on a series of procedural statements and an inference mechanism that mimics human knowledge processing. With respect to the format of the procedural knowledge rules, fuzzy systems may be classified in two types: Mamdani fuzzy systems and Takagi-Sugeno-Kang (TSK) fuzzy systems. Also, there is two major inference mechanisms: composition based inference and individual-rule based inference. For details see [Wang 1997].

In Mandani fuzzy systems, the knowledge rules are of the form:

\[
\text{IF } x_1 \text{ is } X_1' \text{ and } \ldots \text{ and } x_n \text{ is } X_n' \text{,}
\]

\[
\text{THEN } u' \text{ is } U' \text{,} 
\] (5.1)
where the $x_i$, for $i=1,2,\ldots,n$, are the system inputs, and $X_i'$ are fuzzy sets, $u_r$ is the rule output, $U_r$ is an output fuzzy set, and $r=1,2,\ldots,N$, is the rule number index. Both, the antecedent and the consequent of the knowledge rules are fuzzy propositions.

In TSK fuzzy systems, the antecedent of the knowledge rules is a fuzzy proposition, and the consequent is a crisp relation. For first order systems the rule output is calculated as a linear function of the inputs:

$$
IF \ x_i \ is \ X_i' \ and \ \ldots \ and \ x_n \ is \ X_n', \\
THEN \ u^r = c_0^r + c_1^r x_1 + \ldots + c_n^r x_n
$$

(5.2)

where $c_i^r$ are constants. Given input values $x_1, \ldots, x_n$, the total output, $u$, of the TSK fuzzy system is a weighted average of the individual rule outputs:

$$
u = \frac{\sum_r w^r u^r}{\sum_r w^r}
$$

(5.3)

where the weights $w^r$ are calculated as the product of the input membership values:

$$
w_r = \prod_{i=1}^{n} \mu_{X_i'}(x_i)
$$

(5.4)

In retrospective, TSK fuzzy systems are a combination of fuzzy and non-fuzzy models. They integrate qualitative knowledge representation with precise quantitative data expressions. The major advantage of TSK fuzzy systems consists on being universal approximators. They allow the representation of complex nonlinear mappings with simpler linear relations. The knowledge rules set an approximation of a nonlinear input-output mapping, $X_1 \times X_2 \times \cdots \times X_n \rightarrow R$, by a piecewise linear function. The rule antecedents define a decomposition of the input space into a set of overlapping partitions, and establish a switching function that selects, given the actual input values, the appropriate linear functions needed for approximation. Then, the approximated output value is produced by one or interpolated by the combination of two or more relations in the rule consequents, as defined by the inference mechanism of the TSK system.
5.2.2 Feedforward neural-networks

Originally inspired by the neurons in the human brain, artificial neurons are simple signal processing units that can be arranged into large and highly interconnected networks to perform complex information processing tasks such as pattern recognition, data clustering and classification, function approximation, etc. A general model of an artificial neuron (also called a neural unit, or node) considers an input processing function and an output processing function (Figure 5.3).

The input function, \( f(\cdot) \), combines all the input signals into a single net input to the neuron. A common input function is that of a linear combination of the inputs, \( x_i \):

\[
f(x) = \sum_{i=1}^{n} w_i x_i \tag{5.5}
\]

where the \( w_i \), for \( i=1,2,\ldots,n \), are called the link weights. The output function \( g(\cdot) \) provides the output signal of the neuron in terms of the net input value given by \( f \). A common output function is the sigmoid function:

\[
y = g(f) = \frac{1}{1+e^{-f}} \tag{5.6}
\]

An artificial neural network (ANN) is an array of highly interconnected neurons. Among many possible connection geometries, one of the most common and useful, the
multilayer feedforward network (Figure 5.4), in which no neuron output is input to a neuron in the same or in a preceding layer, is of prime interest in this work. The layer of neurons that receives the network inputs is called the input layer. Typically, the input layer does not perform any data processing but only distributes the input signals. The network outputs are generated by the output layer. Any layer between the input and output layers is called a hidden layer because it does not have direct contact with the network environment.

![Figure 5.4 Multilayer feedforward network.](image)

Broadly speaking, there are two kinds of learning in ANNs. Parameter learning is concerned with updating the link weights between neurons, and Structure learning is concerned with the change in network structure, including the number of neurons and the connection geometry. Only parameter learning will be required in this work. In general, parameter-learning rules are classified into three categories: supervised learning, reinforcement learning, and unsupervised learning. Again, only supervised learning is of interest in this work. In gross terms, in supervised learning an ANN is supplied with a sequence of input-output training patterns, then the difference between the actual output produced by the network and the supplied output is successively used to update the
network weights until the difference is minimized. A general training structure for a single neuron is shown in Figure 5.5.

For a given input-output pattern, \((x, y_d)\), the increment of the weighting vector, \(\Delta w\), produced at a learning step time \(k\) is proportional to the product of the input vector to the neuron, \(x\), and a learning signal, \(f_l\), which is a function of the values of the desired neuron output, \(y_d\), and the current actual neuron output, \(y\):

\[
\Delta w_i(k) = \eta f_i(y_d, y(k))x
\]

where \(\eta\) is a positive proportionality constant called the learning constant, which determines the rate of learning, and \(f_i\) is usually a function of the difference between the desired neuron output signal, \(y_d\), and the current actual neuron output, \(y\), which in turn is a function of the current weights, \(w\), and the input vector, \(x\), through the neuron input and output functions. Thus, a general weight-learning rule for a single neuron to fit a single input-output pattern \((x, y_d)\) can be expressed recursively as:

\[
w(k+1) = w(k) + \Delta w(k) = w(k) + \eta f_i(w(k), x, y_d)x
\]
Several weight-learning methods for whole feedforward ANNs have been developed based on (5.8); the main difference among them being the way in which the learning function $f_i$ is implemented. Main interest is on the backpropagation method for multilayer feedforward networks, which is best understood when explained as evolving from the Widrow-Hoff or Adaline learning rule, and the Delta learning rule for single-layer feedforward networks.

The general weight-learning problem for a single-layer feedforward ANN with $m$ inputs, $x_i, i=1,2,\ldots,m$, and $n$ outputs, $y_j, j=1,2,\ldots,n$, is to determine the values of all the link weights $w_k, k=1,2,\ldots,mm$, such that, given a set of $P$ input-output vector training patterns, $\{(x^{(1)}, y_d^{(1)}), (x^{(2)}, y_d^{(2)}), \ldots, (x^{(P)}, y_d^{(P)})\}$, after the learning process the actual network output $y=[y_1 y_2 \cdots y_n]^T$ matches the desired output $y_d=[y_{d1} y_{d2} \cdots y_{dn}]^T$ for all training patterns, that is:

$$y_j^{(p)} = g(w_j^T x^{(p)}) = g\left(\sum_{i=1}^{m} w_{ji} x_i^{(p)}\right) = y_{d_j}^{(p)}, \quad (5.9)$$

where $j=1,2,\ldots,n$, is the output number, $p=1,2,\ldots,P$, is the pattern number, $w_j^T=[w_{j1} w_{j2} \cdots w_{jm}]^T$ is the weight vector associated with the $j$-th neuron, $f_j(x,w_j)=w_j^T x$ is the linear-combination neuron input function of the $j$-th neuron, and $g(\cdot)$ is the neuron output function for any neuron.

The Adaline or Widrow-Hoff learning rule is applied to neurons with a linear input-output relationship known as Adaline (adaptive linear element):

$$y_j^{(p)} = g(f_j) = f_j(w_j, x^{(p)}) = w_j^T x^{(p)} = \sum_{i=1}^{m} w_{ji} x_i^{(p)}, \quad p=1,2,\cdots,P \quad (5.10)$$

for which after learning it is required that $y_j^{(p)} = y_{d_j}^{(p)}$, that is:

$$y_{d_j}^{(p)} = w_j^T x^{(p)} = \sum_{i=1}^{m} w_{ji} x_i^{(p)}, \quad p=1,2,\cdots,P \quad (5.11)$$

which can be viewed as a system of $P$ linear simultaneous equations with $m$ unknowns for each neuron. Usually, the system is overdetermined, i.e., $P > m$, thus instead of an
analytical solution, a solution that minimizes the mean squared error, $E_{ms}(w)$, between the actual outputs and their desired values is preferred:

$$E_{ms}(w) = \frac{1}{2} \sum_{p=1}^{P} \left( y_d^{(p)} - y^{(p)} \right)^2 = \frac{1}{2} \sum_{p=1}^{P} \left( y_d^{(p)} - w^T x^{(p)} \right)^2 = \frac{1}{2} \sum_{p=1}^{P} \left( y_d^{(p)} - \sum_{j=1}^{n} w_j x_j^{(p)} \right)^2$$ (5.12)

which is a monotonic function of the output error. The usual gradient–descent algorithm allows adjusting the weights by an amount proportional and in the opposite direction of the gradient of $E(w)$ at the current location in the weight space:

$$\Delta w = \eta \nabla_w E(w)$$ (5.13)

then for each weight:

$$\Delta w_i = \eta \left. \frac{\partial E(w)}{\partial w_i} \right| = \eta \sum_{p=1}^{P} \left( y_d^{(p)} - w^T x^{(p)} \right) x_j^{(p)}$$ (5.14)

thus the change in the weights due to a single input pattern is:

$$\Delta w_i = \eta \left( y_d^{(p)} - w^T x^{(p)} \right) x_j^{(p)}$$ (5.15)

which is known as the Adaline or Widrow-Hoff learning rule. Comparing (5.15) to the general learning rule (5.8) it can be seen that the learning function for each neuron in this case is:

$$f_i(y_d, w, x) = y_d - w^T x = y_d - y$$ (5.16)

The above learning rule based on the gradient descent technique can be extended to include neurons with differentiable input, $f(\cdot)$, and output, $g(\cdot)$ functions. In this case, the cost function (5.12) becomes:

$$E_{ms}(w) = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{n} \left( y_d^{(p)} - y_j^{(p)} \right)^2 = \frac{1}{2} \sum_{p=1}^{P} \sum_{j=1}^{n} \left( y_d^{(p)} - g \left( f \left( w_j, x_j^{(p)} \right) \right) \right)^2$$ (5.17)

for which:
Hence, the weight correction for a single input pattern is:

$\Delta w_{ji} = -\eta \frac{\partial E_{ms}}{\partial w_{ji}} = -\eta \left( y_{dji}^{(p)} - g \left( f \left( w_{ji}, x_{ji}^{(p)} \right) \right) \right) \frac{\partial g}{\partial f} \frac{\partial f}{\partial w_{ji}}$ (5.19)

which is known as the Delta learning rule. Comparing (5.19) to the general learning rule (5.8) it can be seen that the learning function for each neuron in this case is:

$f_i = \left( y_{dji}^{(p)} - g \left( f \left( w_{ji}, x_{ji}^{(p)} \right) \right) \right) \frac{\partial g}{\partial f}$ (5.20)

So far, the Adaline and Delta learning rules are applied to single layer feedforward networks. The back-propagation learning algorithm (BPLA) extends the results to multilayer feedforward networks with continuously differentiable output functions. In its incremental version, the BPLA updates the network weights once per input-output training pattern considered. For each pattern, the algorithm proceeds in two stages. First, the input pattern $x$ is propagated from the input layer to the output layer, using an arbitrarily given set of weights, to calculate the actual output $y$. Second, the output error with respect to the output pattern $y_d$ is calculated. Then, the error vector is propagated backwards from the output layer to the input layer, updating the weights as passing by through all the network layers. To explain the backpropagation algorithm consider a feedforward network with $m$ input neurons, $n$ output neurons, and $L$ layers, $l=1,2,\ldots,L$. Let $y_{li}$ be the output of the $i$-th neuron in the $l$-th layer, $f_i$ be the input function of the $i$-th neuron in the $l$-th layer, $g$ be the output function of the neurons in the $l$-th layer, and $w_{lj}$ be the connection weight from the $j$-th neuron in the $(l-1)$-th layer to the $i$-th neuron in the $l$-th layer (Figure 5.6). Note that $w_i$ will denote the weight vector of connections from neurons in layer $l-1$ to the $i$-th neuron in layer $l$-th, and $y$ will denote the vector of outputs of neurons in the $l$-th layer.

Once an input pattern has been propagated, the network output error is given by:
where \( L_y = L(g(L_f)) \), and \( L_f = f(L_w, L^{-1}y) \). Then according to the gradient descent method, the weight \( Lw_{ij} \) should be updated by:

\[
\Delta Lw_{ij} = -\eta \frac{\partial E}{\partial Lw_{ij}} = -\eta \frac{\partial E}{\partial L^iy} \frac{\partial L^iy}{\partial L^if} \frac{\partial L^if}{\partial Lw_{ij}}
\]

(5.22)

where for the \( i \)th neuron in layer \( L \):

\[
\Delta Lw_{ij} = \eta \left( y_{di} - L^iy \right) \frac{\partial L^g}{\partial L^if} \frac{\partial L^if}{\partial Lw_{ij}} = \eta L\delta_i \frac{\partial L^f}{\partial Lw_{ij}}
\]

(5.23)

where the error signal \( L\delta_i \) is defined as:

\[
L\delta_i \triangleq -\frac{\partial E}{\partial L^f} = \frac{\partial E}{\partial L^f} \frac{\partial L^g}{\partial L^f} = \left( y_{di} - L^iy \right) \frac{\partial L^g}{\partial L^f}
\]

(5.24)

which constitutes the learning function as in the Delta learning rule.

Next, for a single weight in layer \( L-1 \):
\[ \Delta^{L-1} w_{jk} = -\eta \frac{\partial E}{\partial \Delta^{L-1} w_{jk}} = -\eta \frac{\partial E}{\partial \Delta^{L-1} g_j} \frac{\partial \Delta^{L-1} g_j}{\partial \Delta^{L-1} f_j} \frac{\partial \Delta^{L-1} f_j}{\partial \Delta^{L-1} w_{jk}} \] (5.25)

where \( \Delta^{L-1} g_j \) affects \( E \) throughout all elements of \( \Delta^{L} f \), thus:

\[ \begin{align*}
\Delta^{L-1} w_{jk} &= -\eta \left( \sum_{i=1}^{n} \frac{\partial E}{\partial \Delta^{L} f_j} \frac{\partial \Delta^{L-1} f_j}{\partial \Delta^{L-1} g_j} \right) \frac{\partial \Delta^{L-1} g_j}{\partial \Delta^{L-1} f_j} \frac{\partial \Delta^{L-1} f_j}{\partial \Delta^{L-1} w_{jk}} \\
&= \eta \left( \sum_{i=1}^{n} \Delta_i \frac{\partial \Delta^{L-1} f_j}{\partial \Delta^{L-1} g_j} \right) \frac{\partial \Delta^{L-1} g_j}{\partial \Delta^{L-1} f_j} \frac{\partial \Delta^{L-1} f_j}{\partial \Delta^{L-1} w_{jk}} \\
&= \eta^{L-1} \Delta_i \frac{\partial \Delta^{L-1} f_j}{\partial \Delta^{L-1} w_{jk}}
\end{align*} \] (5.26)

where:

\[ L^{-1} \Delta_i \triangleq \left( \sum_{i=1}^{n} \Delta_i \frac{\partial \Delta^{L} f_j}{\partial \Delta^{L-1} g_j} \right) \frac{\partial \Delta^{L-1} g_j}{\partial \Delta^{L-1} f_j} \] (5.27)

is a recursive algorithm to calculate the error signal (learning function) backwards starting from layer \( L \). Thus (5.26) and (5.27) allow updating the weights in all layers and constitute the backpropagation algorithm.

### 5.2.3 Neuro-fuzzy paradigm

In a hybrid Neuro-Fuzzy System (NFS) a neural network and a fuzzy system are combined into one homogeneous structure. Neuro-fuzzy systems are intended to synthesize the advantages of both neural networks and fuzzy systems in a complementary way to overcome their disadvantages; neural network learning techniques facilitate fuzzy system tuning. There are several approaches for the integration of neural networks and fuzzy systems. The kind of neuro-fuzzy systems of interest for this work have several common defining characteristics. The NFS approximates an \( n \)-dimensional, usually
unknown, function that is partially defined by a set of input-output data. The NFS is a fuzzy system whose knowledge rules represent the relation among samples of the given data. The components of the NFS are determined using neural network learning algorithms applied to the given data. For the purpose of learning, the fuzzy system may be represented by a three-layer feedforward neural network. The first layer represents the input variables, the middle layer represents the fuzzy rules, the third layer represents the output variables, and the connection weights are given specified with fuzzy sets. The neuron units evaluate t-norms and t-conorms operators as activation functions. Models with more than three layers and fuzzy sets as activation functions are also possible. In general, the neural network representation vividly illustrates the parallel nature of fuzzy systems.

The learning procedure is a data-driven process. The learning procedure operates on local information, and causes only local modifications in the underlying fuzzy system. The learning procedure takes into account the properties of the associated fuzzy system, thus constraining the possible modifications of the system’s parameters. Since the NFS is always a fuzzy system at any stage of the learning process, the learning procedure can be initialized specifying the components of a fuzzy system. Consequently, the NFS paradigm may be used to build a fuzzy system from data or to enhance an existing one by learning from examples.

Several methods are currently available to synthesize a neuro-fuzzy system: GARIC, NEFCON, FuNe, ANFIS, etc. [Nauck, et al. 1997]. In this work, the neuro-fuzzy systems are synthesized using the general-purpose adaptive neuro fuzzy inference system (ANFIS) technique [Jang 1993].

5.3 ANFIS Methodology

The ANFIS method allows the design of Sugeno-type fuzzy systems with rules of the form:
where \( j=1,2,\ldots,N \) is the rule number, the \( \textit{LX}^j_i \), for \( i=1,2,\ldots,n \), are the linguistic terms (membership functions) of the input variables \( x_i \), in the antecedent of the \( j \)-th rule, and the \( c_k^j \), for \( k=0,1,\ldots,n \), are the input weighting coefficients to calculate the output \( y^j \) in the consequent of the \( j \)-th rule. Given arbitrary initial knowledge rules, the ANFIS method adjusts the membership functions, \( \textit{LX}^j_i \), and the coefficients, \( c_k^j \), in the consequent of all the rules. To do so, the Sugeno-type fuzzy system is represented as a feedforward neural network, for which its components are refined through standard learning procedures to fit the input-output behavior of the fuzzy system.

5.3.1 Network representation of Sugeno-type fuzzy systems

In the ANFIS method, a fuzzy system with \( n \) inputs and \( N \) rules is represented by a five-layer feedforward network structure with \( N \) neural processing units in layers \( L_1 \), \( L_2 \), \( L_3 \), and \( L_4 \), and 1 unit in \( L_5 \). Layer \( L_0 \) with \( n \) distribution units is not considered a neural processing layer. In gross terms, \( L_1 \) constitutes an input fuzzification stage, then each row across \( L_2 \), \( L_3 \), and \( L_4 \) evaluates a knowledge rule, and finally \( L_5 \) computes the final output value. Neural units in \( L_1 \) and \( L_4 \) are adaptive; their parameters are learned during training. Neural units in \( L_2 \), \( L_3 \), and \( L_5 \) are fixed; their parameters are not modified during training. The exact representation and operation of the network is as follows.

\textit{Layer 1} (\( L_1 \)). Each node receives a single system input and fuzzifies its value, that is, calculates its degree of membership to the linguistic term (fuzzy set) represented by the neural unit. Each node defines a bell-shaped membership function for the associated linguistic term:
\[
\mu (x_i) = \frac{1}{1 + \left( \frac{x_i - c}{a} \right)^2} 
\]

(5.29)

where \( x_i \) is a system input variable, \( a, b, \) and \( c \) are the trainable parameters, and \( \mu(\cdot) \) is the membership value.

\textit{Layer 2 (L2).} Each neural unit in \( L_2 \) is connected to those nodes in \( L_1 \) that form the antecedent of the corresponding rule, thus the inputs to the nodes in \( L_2 \) are degrees of membership. By multiplying all incoming values, each node calculates the degree of fulfillment, \( \tau_r \), of the corresponding rule:

\[
\tau_r = \prod_{i=1}^{n} \mu_r (x_i) \quad r = 1, 2, \cdots, N
\]

(5.30)

\textit{Layer 3 (L3).} Each neural unit in \( L_3 \) is connected to all units in \( L_2 \). Each unit calculates a relative degree of fulfillment of the corresponding rule by normalizing its degree of fulfillment with respect to the degrees of fulfillment of all the rules:

\[
\bar{\tau}_r = \frac{\tau_r}{\sum_{i=1}^{N} \tau_i} 
\]

(5.31)

\textit{Layer 4 (L4).} Each neural unit in \( L_4 \) is connected to one unit in \( L_3 \) and to all the system inputs. Each node calculates the consequent of the corresponding rule weighted by its relative degree of fulfillment:

\[
\bar{y}_r = \bar{\tau}_r y_r = \tau_r (c'_0 x_0 + c'_1 x_1 + c'_2 x_2 + \cdots + c'_n x_n)
\]

(5.32)

where the \( c'_i \), for \( i=0,1,2,\ldots,n \), are trainable parameters.

\textit{Layer 5 (L5).} The only neural unit in \( L_5 \) is connected to all units in \( L_4 \). The node calculates the final output, \( y \), of the fuzzy system by adding all the incoming weighted consequents:
5.3.2 Learning procedure of adaptive components

Given an initial fuzzy system and input-output training data patterns, the ANFIS method allows tuning the parameters of the input membership functions in the nodes of $L_1$ and the coefficients in the knowledge rule consequents in the neural units of $L_4$.

The consequent parameters are estimated using a least square estimation (LSE) procedure. From (5.33) and (5.32) each input-output training pattern can be written as:

$$y = \sum_{i=1}^{N} \tau_i (c_0^i + c_1^i x_1 + c_2^i x_2 + \cdots + c_n^i x_n)$$

(5.34)

where $n$ is the number system inputs, and $N$ the number of rules. Expanding and using vectors:

$$y = [\bar{\tau}_1 \bar{\tau}_1 x_1 \cdots \bar{\tau}_n x_n \cdots \bar{\tau}_N x_1 \cdots \bar{\tau}_N x_n] [c_0^1 c_1^1 \cdots c_0^N c_1^N \cdots c_n^N]^T$$

(5.35)

Considering all $M$ input-output training patterns together:

$$\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_M
\end{bmatrix} =
\begin{bmatrix}
  \bar{\tau}_1 \bar{\tau}_1 x_{11} \cdots \bar{\tau}_1 x_{1n_1} \\
  \bar{\tau}_1 \bar{\tau}_1 x_{12} \cdots \bar{\tau}_1 x_{1n_2} \\
  \vdots \\
  \bar{\tau}_1 \bar{\tau}_1 x_{1M} \cdots \bar{\tau}_1 x_{1nM} \\
  \bar{\tau}_N \bar{\tau}_N x_{N1} \cdots \bar{\tau}_N x_{Nn_1} \\
  \bar{\tau}_N \bar{\tau}_N x_{N2} \cdots \bar{\tau}_N x_{Nn_2} \\
  \vdots \\
  \bar{\tau}_N \bar{\tau}_N x_{N1M} \cdots \bar{\tau}_N x_{NnMM}
\end{bmatrix}
\begin{bmatrix}
  c_0^1 \\
  c_1^1 \\
  \vdots \\
  c_0^N \\
  c_1^N \\
  \vdots \\
  c_n^N
\end{bmatrix}$$

(5.36)

Through appropriate definitions, (5.36) can be written as:

$$Y = XC$$

(5.37)
where \( Y \) is \( M \times 1 \), \( X \) is \( M \times (n+1)N \), and \( C \) is \((n+1)N \times 1\). Usually the problem in (5.37) to calculate the parameters \( C \) is over-determined; there is more patterns than parameters to calculate, \( M > (n+1)N \), and there is no exact solution. Instead, a least squares estimate of \( C \) that minimizes the squared error \( \| X C - Y \|^2 \) is obtained:

\[
C_{lse} = (X^T X)^{-1} X Y
\]  

(5.38)

where \((X^T X)^{-1}\) is the pseudo-inverse of \( X \) if \( X^T X \) is nonsingular. The solution in (5.38) is computationally expensive due to the matrix inversion, or impossible if \( X^T X \) is singular. To avoid these problems, a more efficient recursive procedure is used. Let \( \bar{x}_i \) be the \( i \)th row vector of matrix \( X \) and \( y_i \) the \( i \)th element of vector \( Y \). Then a LSE solution for \( C \) can be computed recursively using the widely adopted method [Goodwin and Sin 1984]:

\[
C_{i+1} = C_i + \Psi_{i+1} x_i (y_i - x_i^T C_i)
\]  

(5.39)

\[
\Psi_{i+1} = \Psi_i - x_i^T x_{i+1} \Psi_i x_i^T (1 + x_i^T \Psi_i x_{i+1})^{-1}
\]  

(5.40)

for \( i=0,1,2,\ldots,M-1 \), then \( C_{lse}=C_M \), where \( \Psi \) is called the covariance matrix. The initial conditions are \( C_0 = 0 \) and \( \Psi_0 = \gamma I \), where \( I \) is a size \((n+1)N\) identity matrix and \( \gamma \) is a large positive number.

The changes to the antecedent parameters are determined by backpropagation. Let \( z \) be any of the \( a, b, \) or \( c \) parameters of any membership functions \( \mu \), and \( E \) be the usual error measure (5.21) given by the sum of squared difference between the target output, \( y^* \), and actual output, \( y \). Then the change in \( z \), \( \Delta z \), for a single rule after a pattern has been propagated is given by:

\[
\Delta z = -\sigma \frac{\partial E}{\partial z}
\]  

(5.41)

where \( \sigma \) is an arbitrary learning rate factor. Successive application of the chain rule in (5.41) yields:
\[
\Delta z = -\sigma \frac{\partial E}{\partial y} \frac{\partial y}{\partial \tau} \frac{\partial \tau}{\partial r} \frac{\partial r}{\partial \mu} \frac{\partial \mu}{\partial z}
\]

\[
= \sigma (y^* - y) \frac{\tau_r (1 - \tau_r)}{\tau_r} \frac{\partial \mu}{\partial z}
\]

\[
= \frac{\sigma}{\mu} y_r (y^* - y) \tau_r (1 - \tau_r) \frac{\partial \mu}{\partial z}
\]

where the last factor depends on the membership function parameter being considered:

\[
\frac{\partial \mu}{\partial a} = \mu(x_i)^2 \left( \frac{(x_i - c)^2}{a} \right)^b
\]

\[
\frac{\partial \mu}{\partial b} = -b \mu(x_i)^2 \left( \frac{(x_i - c)^2}{a} \right)^{(b-1)}
\]

\[
\frac{\partial \mu}{\partial c} = 2b \mu(x_i)^2 \left( \frac{(x_i - c)^2}{a} \right)^b
\]

The learning process is carried out iteratively, with two phases per iteration. First, the input patterns are propagated keeping the antecedent parameters constant, then the optimal consequent parameters are estimated using the least square estimation procedure in (5.39) and (5.40). Second, the input patterns are propagated again while keeping the consequent parameters constant, then the antecedent parameters are modified by backpropagation using (5.42) through (5.45).

5.4 Feedforward Control Processor

5.4.1 Neuro-fuzzy feedforward controller

As mentioned earlier, the feedforward controller in the ICCS-MP consists of three MISO fuzzy systems that provide the feedforward control signals for the fuel, \( u_{1fg} \), steam,
and feedwater, \( u_{2ff} \), and control valves, in terms of the power, \( E_d \), pressure, \( P_d \), and level, \( L_d \), set-points, as was shown in Figure 5.2. The feedforward controller design problem may be stated as: Given a set of steady-state input-output patterns, \([u_1 \ u_2 \ u_3 \ E \ P \ L] \), determine the MISO fuzzy systems FISU1, FISU2, and FISU3. More specifically, the problem consists in finding out the values of the parameters of the membership functions in the rule antecedents and the coefficients in the rule consequents of the three TSK-type fuzzy systems, so that the set of inverse steady-state input-output patterns, \([E \ P \ L \ u_1 \ u_2 \ u_3] \), are matched. Note that FISU1, FISU2, and FISU3 should reproduce the sets of patterns: \([E \ P \ L \ u_1] \), \([E \ P \ L \ u_2] \), and \([E \ P \ L \ u_3] \) as \([E_d \ P_d \ L_d \ u_{1ff}] \), \([E_d \ P_d \ L_d \ u_{2ff}] \), and \([E_d \ P_d \ L_d \ u_{3ff}] \), respectively, once embedded in the ICCS-MP. The feedforward controller design problem is solved independently for each fuzzy system using the necessary data from the same set of steady-state input-output patterns, \([u_1 \ u_2 \ u_3 \ E \ P \ L] \).

All three fuzzy systems are of the TSK-type and have similar structures, so without loss of generality and to simplify the presentation, and unless otherwise specified, hereafter all explanations refer to FISU1, the system that generates \( u_{1ff} \). The knowledge rules of the fuzzy system have the form:

\[
\text{IF: } E \text{ is LE}_d' \text{ and } P \text{ is LP}_d' \text{ and } L \text{ is LL}_d'
\]

\[
\text{THEN: } u_{1ff}^r = c_0^r + c_{E}^r E_d + c_{P}^r P_d + c_{L}^r L_d
\]

(5.46)

where \( r=1,2,\ldots, R \) is the rule number, \( \text{LE}_d' \), \( \text{LP}_d' \), and \( \text{LL}_d' \) are the linguistic terms of the input signals \( E_d, P_d, \) and \( L_d \), respectively, in the \( r \)-th rule, \( u_{1ff}^r \) is the contribution of the \( r \)-th rule to the total output of the fuzzy system, and \( c_0^r, c_{E}^r, c_{P}^r, \) and \( c_{L}^r \) are the consequent coefficients. For a given input pattern \([E_d \ P_d \ L_d]^{\top} \), the output of the fuzzy system is given by:

\[
u_{1ff} = \frac{\sum_{r=1}^{R} w^r u_{1ff}^r}{\sum_{r=1}^{R} w^r}
\]

(5.47)
where $w_r$, for $r = 1, 2, \cdots, R$, are the rule fulfillment weights. For each rule, its weight is calculated as the product of the input membership values as:

$$w_r = \mu_{LE_d}(E_d) \cdot \mu_{LP_d}(P_d) \cdot \mu_{LL_d}(L_d)$$  \hspace{1cm} (5.48)

where $\mu_{LE_d} (\cdot)$, $\mu_{LP_d} (\cdot)$, and $\mu_{LL_d} (\cdot)$ are the membership functions corresponding to the linguistic terms $LE_d$, $LP_d$, and $LL_d$, respectively, in the $r$-th rule. In addition, note that (5.47) can be written as:

$$u_{ij} = \frac{\sum_{r=1}^{R} w_r}{\sum_{r=1}^{R} w_r} \cdot \bar{w}'_r \cdot u_{ij}' = \sum_{r=1}^{R} \bar{w}'_r \cdot u_{ij}'$$  \hspace{1cm} (5.49)

where $\bar{w}'_r$, for $r = 1, 2, \cdots, R$, are the so called (normalized) relative rule fulfillment weights:

$$\bar{w}'_r = \left( \frac{w_r}{\sum_{r=1}^{R} w_r} \right)$$  \hspace{1cm} (5.50)

and $\bar{u}_{ij}'$, for $r = 1, 2, \cdots$, can be equivalently called the normalized rule consequents:

$$\bar{u}_{ij}' \triangleq \bar{w}'_r \cdot u_{ij}'$$  \hspace{1cm} (5.51)

Each fuzzy system is to be designed with the previously explained ANFIS technique. To this aim, the fuzzy system is represented as a 3-input-1-output 5-layer feedforward neural network, as shown in Figure 5.7 for the case where, without lose of generality, each input signal spans its whole operating range with three overlapping fuzzy regions, that is using three fuzzy sets with bell-shaped membership functions and linguistic terms: low, medium, and high. Therefore, for this case a complete knowledge base will have $3 \times 3 \times 3 = 27$ rules of the form given in (5.46) Also, the network will have 3 distribution units in layer $L_0$, 9 neurons in $L_1$, 27 neurons in $L_2$, $L_3$, and $L_4$, and 1 neuron in $L_5$. With these dimensions, the number of parameters to determine is calculated as
follows: 27 rules × 4 consequent parameters per rule = 108 consequent parameters, and 3 inputs × 3 membership functions per input × 3 parameters per membership function = 27 membership function parameters. Then, the total number of parameters to be determined is 108 + 27 = 135 per fuzzy system. This numbers clearly illustrate the difficulty of tuning a fuzzy system following a trial & error approach, which simply gets worse as the number of input linguistic terms increases. Fortunately, this process can be fully automated using the neuro-fuzzy paradigm and a low-dimension fuzzy system will perfectly do the job, as will be shortly shown.

Figure 5.7 Neural network structure of feedforward fuzzy controller.
Each neuron in layer \( L_1 \) fuzzifies the incoming input signal using bell-shaped membership functions. In this layer, the neuron’s input and output processing functions are of the form:

\[
^1f_i(x) = x \tag{5.52}
\]

\[
^1y_i = ^1g_i(f) = ^1g_i(x) = \frac{1}{1 + \left( \frac{x - c_i}{a_i} \right)^{2b_i}} \tag{5.53}
\]

where \( i = 1, 2, \ldots, 9 \) is the neuron number, \( x = E_d, P_d, L_d \) is the input signal, \(^1y_i\) is the \( i\)-th neuron output, and \( a_i, b_i, \) and \( c_i \) are the parameters of the bell-shaped output function.

Neurons in layer \( L_2 \) calculate the rule fulfillment weight for each rule using the following input and output processing functions:

\[
^2f_i(^1x_{i1}, ^1x_{i2}, ^1x_{i3}) = ^1x_{i1} \cdot ^1x_{i2} \cdot ^1x_{i3} \tag{5.54}
\]

\[
w_r = ^2y_i = ^2g_i(^2f_i) = ^1x_{i1} \cdot ^1x_{i2} \cdot ^1x_{i3} \tag{5.55}
\]

where \( i = 1, 2, \ldots, 27 \) is the neuron number, \(^1x_{i1}, ^1x_{i2}, \) and \(^1x_{i3} \) are the inputs to the \( i\)-th neuron, \(^2y_i\) is the output of the \( i\)-th neuron, and \( w_r \) is the rule fulfillment weight of the \( r\)-th rule with \( r=i \).

Neurons in layer \( L_3 \) calculate the relative rule fulfillment weight for each rule through:

\[
^3f_i(^3x_{i1}, ^3x_{i2}, \cdots, ^3x_{i27}) = ^3x_{i1} + ^3x_{i2} + \cdots + ^3x_{i27} = \sum_{j=1}^{27} ^3x_{ij} = \sum_{r=1}^{27} w_r \tag{5.56}
\]

\[
\bar{w}_r = ^3y_i = ^3g_i(^3f_i) = \frac{^3x_{ir}}{\sum_{r=1}^{27} w_r} \tag{5.57}
\]

where \( i = 1, 2, \ldots, 27 \) is the neuron number, \(^3x_{ij}\) for \( j = 1, 2, \ldots, 27 \) are the inputs to the \( i\)-th neuron coming from all the neuron outputs in \( L_2 \), \(^3y_i\) is the output of the \( i\)-th neuron, and \( \bar{w}_r \) is the (normalized) relative rule fulfillment weight of the \( r\)-th rule with \( r=i \).
Neurons in layer \( L_4 \) calculate the normalized rule consequent for each rule through:

\[
4f_i(x_i) = 4x_i = \bar{w}_r
\]  

(5.58)

\[
\bar{u}_r = 4y_i = 4g_i(4f_i, 0x_1, 0x_2, 0x_3)
\]

\[
= 4f_i(c_{r0} + c_{r1} \cdot 0x_1 + c_{r2} \cdot 0x_2 + c_{r3} \cdot 0x_3)
\]

(5.59)

\[
= \bar{w}_r(c_{r0} + c_{r1} \cdot E_d + c_{r2} \cdot P_d + c_{r3} \cdot L_d)
\]

where \( i=1,2,\ldots,27 \) is the neuron number, \( 4x_i \) is the input to the \( i-th \) neuron, \( 4y_i \) is the output of the \( i-th \) neuron, and \( \bar{w}_r \) is the normalized rule consequent of the \( r-th \) rule with \( r=i \).

The unique neuron in layer \( L_5 \) calculates the total system output via:

\[
5f_1(x_1, x_2, \ldots, x_{27}) = 5x_1 + 5x_2 + \cdots + 5x_{27} = \sum_{r=1}^{27} \bar{u}_r
\]  

(5.60)

\[
u_{1ff} = 5y_i = 5g_1(5f_1) = \sum_{r=1}^{27} \bar{u}_r
\]  

(5.61)

where \( u_{1ff} \) is the total output of the FISU1 system used in this description.

### 5.4.2 Neural-network supervisory designer

As for the neuro-fuzzy feedforward controller, the description of the supervisory design process is given for the FISU1 system for the case with three linguistic terms per input, that is with 27 knowledge rules. An analogous description applies to the other fuzzy systems.

Given a set of \( M \) steady-state input-output patterns \( \{[u_{11} u_{21} u_{31} E_1 P_1 L_1], \ldots, [u_{1M} u_{2M} u_{3M} E_M P_M L_M]\} \), and an initial MISO TSK fuzzy system defined as in (5.46)-(5.51) and specified by arbitrary sets of parameters \( \{[a_1 b_1 c_1], \ldots, [a_9 b_9 c_9]\} \) and \( \{[c_{O1} c_{E1} c_{P1} c_{L1}], \ldots, [c_{O27} c_{E27} c_{P27} c_{L27}]\} \) corresponding to the membership functions and the consequent coefficients, respectively; the supervisory designer adjusts the parameters of
FISU1 so that it reproduces the set of patterns \([E_1 P_1 L_1 u_{11}], \ldots, [E_M P_M L_M u_{1M}]\) corresponding to the inverse static model to generate \(u_{1ff}\).

The consequent parameters are to be estimated using a LSE procedure. From (5.61) and (5.59), each input-output pattern is related by:

\[
u_i = \sum_{r=1}^{27} \bar{w}_r \left(c_{or} + c_{Er}E_i + c_{Pr}P_i + c_{Lr}L_i \right)
\]

(5.62)

where \(u\) was used instead of \(u_{1ff}\) to simplify the notation. Using vectors and considering all \(M\) input-output training patterns:

\[
\begin{bmatrix}
u_1 \\
\vdots \\
u_M
\end{bmatrix} = 
\begin{bmatrix}
\bar{w}_1 \bar{w}_1 E_1 \bar{w}_1 P_1 \bar{w}_1 L_1 \\
\vdots \\
\bar{w}_{27} \bar{w}_{27} E_{27} \bar{w}_{27} P_{27} \bar{w}_{27} L_{27}
\end{bmatrix}
\begin{bmatrix}
c_{o1} \\
c_{E1} \\
c_{P1} \\
c_{L1} \\
\vdots \\
c_{o27} \\
c_{E27} \\
c_{P27} \\
c_{L27}
\end{bmatrix}
\]

(5.63)

which through appropriate definitions can be written as:

\[
U = XC
\]

(5.64)

where \(U\) is \(M\times1\), \(X\) is \(M\times(4)(27)=M\times108\), and \(C\) is \(108\times1\). In general the problem is overdetermined, that is \(M>108\). An LSE solution for \(C\) can be computed recursively using (5.39) and (5.40) as:

\[
C_{i+1} = C_i + \Psi_{i+1} x_{i+1} (u_{i+1}^T - x_{i+1}^T C_i)
\]

(5.65)

\[
\Psi_{i+1} = \Psi_i - \frac{\Psi_i x_{i+1}^T x_{i+1} \Psi_i}{1 + x_{i+1}^T \Psi_i x_{i+1}}
\]

(5.66)

where \(x_i^T\) is the \(i\)th row vector of matrix \(X\) and \(u_i\) is the \(i\)th element of vector \(U\), for \(i=0, 1, 2, \ldots, M-1\), and \(\Psi\) is the covariance matrix. The initial conditions are \(C_0 = 0\) and \(\Psi_0 = \gamma I\).
where \( I \) is a size 108 identity matrix and \( \gamma \) is a large positive number. At the end of iterations: \( C = C_M \).

Complementarily, the changes to the membership function parameters are determined by backpropagation. The parameter changes for a single rule after a pattern has been propagated are calculated using the relationships (5.41) through (5.45) yielding:

\[
\Delta a = \frac{\sigma}{\mu} u_r (u^* - u) \overline{w}_r (1 - \overline{w}_r) \frac{\mu^2}{a} \left( \frac{(E - c)^2}{a} \right)^b \tag{5.67}
\]

\[
\Delta b = -\frac{\sigma}{\mu} u_r (u^* - u) \overline{w}_r (1 - \overline{w}_r) b \mu^2 (E) \left( \frac{(E - c)^2}{a} \right)^{(b-1)} \tag{5.68}
\]

\[
\Delta c = \frac{\sigma}{\mu} u_r (u^* - u) \overline{w}_r (1 - \overline{w}_r) \frac{2b \mu^2 (E)}{E - c} \left( \frac{(E - c)^2}{a} \right)^b \tag{5.69}
\]

As previously mentioned the learning process is carried out iteratively. The procedure consist of the following steps:

1) Propagate all patterns from the training set and determine the consequent parameters by the iterative LSE in (5.65) and (5.66). During this step the antecedent parameters remain fixed.

2) Propagate all patterns again and update the antecedent parameters by backpropagation using (5.67)-(5.69). During this step the consequent parameters remain fixed.

3) If the error is reduced in four consecutive steps then increase the learning rate by 10%. If the error is subject to consecutive combinations of increase and reduction, then decrease the learning rate by 10%.

4) Stop if the error is small enough, otherwise continue with step 1).

For practical application, the learning process is incorporated in a three stage design process. First, a set of input-output data, to be used as training data, needs to be generated or obtained from the process. Another optional data set can be used as
checking data after training to evaluate the performance of the learning process. Second, initial structures for the FIS need to be created. For each input, the range of operation, number of membership functions, as well as their shape, must be defined. Finally, the previously defined learning process is carried out using the training data set to adjust the membership functions, and to determine the consequent parameters. The resultant FIS is verified using the checking data set.

5.5 Design of Neuro-Fuzzy Controllers

In this section the supervisory designer is used to implement the neuro-fuzzy controllers in the ICCS-MP. Some results are provided to demonstrate the feasibility of the proposed approach to design the feedforward controller and to show its main characteristics. First, the data of four cases of interest to design the controllers is presented. Second, the effect of number of membership functions and number of training epochs on the approximation accuracy of the neuro-fuzzy controller to the inverse static model of the FFPU is illustrated for the sliding pressure case. Finally, the design of the neuro-fuzzy controllers is presented for all four cases.

5.5.1 Case studies

The four cases of interest to illustrate the design of the neuro-fuzzy feedforward controller include the constant-pressure operating policy defined in Table 2.6, the sliding-pressure operating policy defined in Table 2.7, the upper and lower limits of the power-pressure operating region shown in Figure 2.9. These cases constitute a reasonable good sample of power-pressure policies covering the whole operating window of the FFPU. All necessary steady-state input-output data to design the controllers is obtained from these sources. Recall that since the controllers are to approximate the inverse static
behavior of the FFPU, the control signals will be considered as outputs and the power, pressure and drum water level deviation will be the inputs. Figure 5.8 shows the data for the pressure, $P$, and the control signals $u_1$, $u_2$, and $u_3$, for the upper-limit pressure case with power $E$ as independent variable. Note that the drum water level deviation $L$ is not shown since in steady-state it is always zero for all loads. Figure 5.9 provides the steady-state input-output data for the constant-pressure case. Figure 5.10 provides the steady-state input-output data for the sliding-pressure case. Finally, Figure 5.11 comprehends the lower-limit pressure case.

Figure 5.8 Input-output steady-state data for upper-pressure limit case.
Figure 5.9 Input-output steady-state data for constant pressure case.

Figure 5.10 Input-output steady-state data for sliding pressure case.
5.5.2 Effect of membership function number and training epochs

Assuming that the data required to design the neuro-fuzzy controllers is already available, two major decisions have to be made in order to obtain controllers with satisfactory performance. First, the number of linguistic terms (equivalently the number of membership functions) to be used to fuzzify the input signals has to be decided. The number of linguistic terms per input not only determines the size of the knowledge base, that is, the number of knowledge rules, but also will determine the number of parameters to be calculated, and the number of input-output data patterns required for the learning process. Second, it must be decided how to stop the learning process. The stopping condition may be set in terms of reaching a predefined approximation accuracy, or in terms of the execution of a predefined number of training iterations (epochs). Whatever is decided, the effect of major interest is the impact on the accuracy of the resulting fuzzy

Figure 5.11 Control signal steady-state values for the lower-pressure limit case.
system to approximate the set of data patterns provided for learning the fuzzy system components, that is, the main concern is with the accuracy of the inverse steady-state model of the power unit. In any case the performance is evaluated as a measure of the output approximation error. Ideally, it will always be preferred a low dimensional system requiring a small number of training iterations if the obtained approximation accuracy is reasonably close to that of a high dimensional system requiring a large number of training iterations.

In what follows the effect of both the number of linguistic terms and the number of learning iterations on the approximation accuracy is shown. Note that since all three neuro-fuzzy controllers exhibited similar characteristics, only the results for FISU1 are provided. Later, in Section 5.5 final design results are provided for all three controllers. First, the effect of the number of membership functions is illustrated for the sliding-pressure case using 3, 5, and 7 membership functions. For each sub-case, the root squared mean error of the output approximation (RSME) and the resultant membership functions are plotted in Figures 5.12 through 5.15. Note that the same number of linguistic terms is used for every input of the neuro-fuzzy controller. Also, only the membership functions for the power and pressure inputs are provided since, for all cases, the membership functions of the drum water-level deviation are singletons at $L=0$.

![Figure 5.12 RSME approximation accuracy for sliding-pressure case.](image)
Figure 5.13  FISU1 membership functions, sliding-pressure, 3 linguistic terms.

Figure 5.14  FISU1 membership functions, sliding-pressure, 5 linguistic terms.
The effect of the number of training epochs is illustrated for the sliding-pressure case with 3 input membership functions; the root squared mean error and the resulting membership functions are plotted for training during 20 epochs in Figures 5.16 and 5.17.

By inspection of these results, it can be seen that very good approximations of the inverse steady-state model of the plant can be obtained with low-dimension systems and small number of training epochs. This is due to the fact that the nonlinear steady-state behavior of the plant is benign, that is the non-linearities are smooth and continuous.
Figure 5.16 RSME for sliding-pressure case, 3, 5 and 7 mf, 20 epochs.

Figure 5.17 FISU1 membership functions for sliding-pressure case, 3 mf, 20 epochs.
5.5.3 Feedforward fuzzy systems

With the observations made in the last section, what rests to demonstrate is that the three FIS in the ICCS-MP can be satisfactorily constructed anywhere in the power-pressure operating window of the unit. To this aim, the design of FISU2 and FISU3 is presented along with the already shown FISU1 for the four cases being considered: high-pressure limit (HP), constant pressure (CP), sliding pressure (SP), and low-pressure limit (LP). In all cases, 3 membership functions and 10 training epochs were considered. First, Table 5.1 summarizes the approximation performance of each FIS in generating the corresponding steady-state feedforward control signals. Second, Figures 5.18 through 5.23 show the resultant membership functions for the power and pressure inputs. Then, Figures 5.24 through 5.26 show the fuzzy surfaces over the power-pressure plane for each FIS for the four power-pressure relationships being considered. Note that each FIS is graphically represented by a fuzzy inference surface, which is a more intuitive and abbreviated representation than the corresponding knowledge base. This can be appreciated comparing Figure 5.24c to Table 5.2, both representing FISU1 for the sling pressure case. Also, the graphical representation has the advantage that it changes very little with increasing number of membership functions, but the table will exhibit the curse of dimensionality problem (i.e., the number of rules increases geometrically with the number of membership functions).

<table>
<thead>
<tr>
<th>Output</th>
<th>HP</th>
<th>CP</th>
<th>SP</th>
<th>LP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_1^{ff} )</td>
<td>8.3013</td>
<td>9.8344</td>
<td>8.5579</td>
<td>3.8310</td>
</tr>
<tr>
<td>( u_2^{ff} )</td>
<td>69.8810</td>
<td>9.4715</td>
<td>265.2692</td>
<td>129.9016</td>
</tr>
<tr>
<td>( u_3^{ff} )</td>
<td>10.5651</td>
<td>10.4077</td>
<td>24.6003</td>
<td>20.9564</td>
</tr>
</tbody>
</table>
Figure 5.18 FISU1 power membership functions.

Figure 5.19 FISU1 pressure membership functions.
Figure 5.20  FISU2 power membership functions.

Figure 5.21  FISU2 pressure membership functions.
Figure 5.22 FISU3 power membership functions.

Figure 5.23 FISU3 pressure membership functions.
Figure 5.24 FISU1 fuzzy inference surfaces.
Figure 5.25 FISU2 fuzzy inference surfaces.
Figure 5.26 FISU3 fuzzy inference surfaces.
<table>
<thead>
<tr>
<th>#</th>
<th>$E_d$</th>
<th>$P_d$</th>
<th>$L_d$</th>
<th>$c_E$</th>
<th>$c_P$</th>
<th>$c_L$</th>
<th>$c_I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mf1</td>
<td>mf1</td>
<td>mf1</td>
<td>1.5291</td>
<td>-1.5612</td>
<td>0.0000</td>
<td>-0.0041</td>
</tr>
<tr>
<td>2</td>
<td>mf1</td>
<td>mf1</td>
<td>mf2</td>
<td>19.9689</td>
<td>-20.5440</td>
<td>0.0000</td>
<td>-0.0543</td>
</tr>
<tr>
<td>3</td>
<td>mf1</td>
<td>mf1</td>
<td>mf3</td>
<td>1.5470</td>
<td>-1.5877</td>
<td>0.0000</td>
<td>-0.0042</td>
</tr>
<tr>
<td>4</td>
<td>mf1</td>
<td>mf2</td>
<td>mf1</td>
<td>0.1928</td>
<td>0.3392</td>
<td>0.0000</td>
<td>-0.0229</td>
</tr>
<tr>
<td>5</td>
<td>mf1</td>
<td>mf2</td>
<td>mf2</td>
<td>1.7895</td>
<td>2.5991</td>
<td>0.0000</td>
<td>-0.3089</td>
</tr>
<tr>
<td>6</td>
<td>mf1</td>
<td>mf2</td>
<td>mf3</td>
<td>0.0869</td>
<td>-0.0084</td>
<td>0.0000</td>
<td>-0.0249</td>
</tr>
<tr>
<td>7</td>
<td>mf1</td>
<td>mf3</td>
<td>mf1</td>
<td>0.3965</td>
<td>0.0948</td>
<td>0.0000</td>
<td>-0.0579</td>
</tr>
<tr>
<td>8</td>
<td>mf1</td>
<td>mf3</td>
<td>mf2</td>
<td>5.1886</td>
<td>1.3658</td>
<td>0.0000</td>
<td>-0.7574</td>
</tr>
<tr>
<td>9</td>
<td>mf1</td>
<td>mf3</td>
<td>mf3</td>
<td>0.3955</td>
<td>0.1010</td>
<td>0.0000</td>
<td>-0.0587</td>
</tr>
<tr>
<td>10</td>
<td>mf2</td>
<td>mf1</td>
<td>mf1</td>
<td>0.6747</td>
<td>0.4897</td>
<td>0.0000</td>
<td>0.0084</td>
</tr>
<tr>
<td>11</td>
<td>mf2</td>
<td>mf1</td>
<td>mf2</td>
<td>5.0558</td>
<td>1.1478</td>
<td>0.0000</td>
<td>0.0802</td>
</tr>
<tr>
<td>12</td>
<td>mf2</td>
<td>mf1</td>
<td>mf3</td>
<td>0.2088</td>
<td>-0.0783</td>
<td>0.0000</td>
<td>0.0051</td>
</tr>
<tr>
<td>13</td>
<td>mf2</td>
<td>mf2</td>
<td>mf1</td>
<td>0.9172</td>
<td>1.1648</td>
<td>0.0000</td>
<td>0.0202</td>
</tr>
<tr>
<td>14</td>
<td>mf2</td>
<td>mf2</td>
<td>mf2</td>
<td>4.8350</td>
<td>1.2250</td>
<td>0.0000</td>
<td>0.1920</td>
</tr>
<tr>
<td>15</td>
<td>mf2</td>
<td>mf2</td>
<td>mf3</td>
<td>1.4916</td>
<td>1.6717</td>
<td>0.0000</td>
<td>0.0238</td>
</tr>
<tr>
<td>16</td>
<td>mf2</td>
<td>mf3</td>
<td>mf1</td>
<td>0.6634</td>
<td>0.1583</td>
<td>0.0000</td>
<td>-0.0188</td>
</tr>
<tr>
<td>17</td>
<td>mf2</td>
<td>mf3</td>
<td>mf2</td>
<td>7.9295</td>
<td>0.1758</td>
<td>0.0000</td>
<td>-0.2560</td>
</tr>
<tr>
<td>18</td>
<td>mf2</td>
<td>mf3</td>
<td>mf3</td>
<td>0.5596</td>
<td>-0.1891</td>
<td>0.0000</td>
<td>-0.0208</td>
</tr>
<tr>
<td>19</td>
<td>mf3</td>
<td>mf1</td>
<td>mf1</td>
<td>0.4535</td>
<td>0.1746</td>
<td>0.0000</td>
<td>-0.0423</td>
</tr>
<tr>
<td>20</td>
<td>mf3</td>
<td>mf1</td>
<td>mf2</td>
<td>5.2851</td>
<td>1.2473</td>
<td>0.0000</td>
<td>-0.5593</td>
</tr>
<tr>
<td>21</td>
<td>mf3</td>
<td>mf1</td>
<td>mf3</td>
<td>0.3579</td>
<td>0.0199</td>
<td>0.0000</td>
<td>-0.0437</td>
</tr>
<tr>
<td>22</td>
<td>mf3</td>
<td>mf2</td>
<td>mf1</td>
<td>0.4854</td>
<td>0.5503</td>
<td>0.0000</td>
<td>0.0062</td>
</tr>
<tr>
<td>23</td>
<td>mf3</td>
<td>mf2</td>
<td>mf2</td>
<td>3.3981</td>
<td>3.1480</td>
<td>0.0000</td>
<td>0.0579</td>
</tr>
<tr>
<td>24</td>
<td>mf3</td>
<td>mf2</td>
<td>mf3</td>
<td>0.1272</td>
<td>0.1309</td>
<td>0.0000</td>
<td>0.0037</td>
</tr>
<tr>
<td>25</td>
<td>mf3</td>
<td>mf3</td>
<td>mf1</td>
<td>0.8719</td>
<td>1.0047</td>
<td>0.0000</td>
<td>0.0052</td>
</tr>
<tr>
<td>26</td>
<td>mf3</td>
<td>mf3</td>
<td>mf2</td>
<td>11.3758</td>
<td>13.0895</td>
<td>0.0000</td>
<td>0.0677</td>
</tr>
<tr>
<td>27</td>
<td>mf3</td>
<td>mf3</td>
<td>mf3</td>
<td>0.8811</td>
<td>1.0142</td>
<td>0.0000</td>
<td>0.0052</td>
</tr>
</tbody>
</table>
5.6 Summary

This chapter presented the design of the feedforward control processor module of the ICCS-MP. The module is realized as a supervisory designer and a feedforward controller that approximates the inverse steady-state behavior of the unit along its whole operating range through a set of independent nonlinear multivariable mappings that calculate the feedforward control signals from the control system set-points. Each mapping is a MISO fuzzy inference of the TSK type that is designed from input-output data patterns by the supervisory designer using a neuro-fuzzy paradigm. This approach provides off-line operator requested system adaptability that can be used to achieve optimal operation, and the structure of the module allows the control structure to be easily expanded or contracted, thus supporting the whole system openness characteristic.
CHAPTER 6

FEEDBACK CONTROL PROCESSOR

In this chapter, the feedback control processor module of the ICCS-MP is described. In Section 6.1, the proposed approach is briefly justified and the feedback control module is overviewed. Section 6.2 presents some remarks on multiloop decoupling control and describes the design of the interaction compensator component of the feedback control. Section 6.3 presents some remarks on gain scheduling and multimode control, and introduces the basic fuzzy-PID controller component of the feedback control. In Section 6.4, the supervisory tuning strategy based on the simple genetic algorithm is presented. In Section 6.5, the design of the 2D scheduling strategy for both the controllers and compensator is described in detail. Finally, Section 6.6 summarizes this chapter. Because of its nature that requires all the components of the control system to be already in place, the treatment of the supervisory controller tuner is postponed to the next chapter, where the ICCS-MP is integrated.

6.1 Introduction

Most of current control systems for fossil-fuel power plants basically consist of multiloop configurations of SISO feedback control loops complemented with many feedforward compensations. The control loops normally evaluate conventional PI or PID algorithms, and the feedforward compensation functions have been added to minimize the effects of interaction among the control loops. Many different control system configurations have been developed after various decades of research and practical experience. Their effectiveness to regulate the process under random load disturbances is daily proven all around the world. However, current requirements demanding wide-range
operation of fossil-fuel units seriously challenge most of these configurations, which are better suited for operation around a single fixed operating point. Normally, the parameters of the controllers and compensation functions are tuned at some predefined operating point \((i.e.,\) base load) assuming nearly constant load conditions, and left fixed thereafter. Consequently, the performance of the power unit may decrease due to the nonlinear and interactive dynamics of the process, that change with the actual operating point. Furthermore, due to the mismatch between the tuning point and the actual operating point, the plant equipment may be subject to strong physical demands, which can be very detrimental of the unit’s duty life. Therefore, it would be highly desirable to extend the operating range of a control strategy, of already proven effectiveness, keeping the same structure and with controllers of the same generic form. This approach may significantly improve the plant performance, while minimizing the impact on current operation procedures; maintenance, operation, and fuel costs may be favorably affected.

Gain scheduling and multi-mode control have been pointed out as two effective strategies for controlling nonlinear processes over wide operating regions. In the gain scheduling approach a controller is designed using linear techniques at several operating points that are defined in terms of a scheduling variable, which is representative of the current process operating conditions. Then, during operation the controller parameters are interpolated between the values at the selected operating points. In the multi-mode approach a set of controllers, not necessarily of the same class, is designed for different operating conditions or phases. During operation, all controllers are executed in parallel, and a supervisor selects the controller to be used depending on the current operating conditions. In this research, both approaches are combined into a single strategy using fuzzy systems to design a new class of PID-like controllers, adequate for wide-range operation, that can be used directly in multiloop control configurations instead of the conventional PID controllers.

Unfortunately, most feedforward compensation structures are not designed in a systematic way; their design still heavily depends on the designers’ experience. Also, there are no compensation methods for wide-range operation available in the technical
literature. To overcome these problems a systematic method to design the feedforward compensation structure in the form of an interaction compensator is presented, together with a scheduling strategy that extends the application of the compensator to the whole operating window of the unit.

In this chapter, the feedback control processor (FBCP) module of the ICCS-MP is presented (Figure 6.1). The purpose of the FBCP is to provide corrective control actions along the commanded set-point trajectories to overcome the effect of disturbances and uncertainties in the whole operating window of the power unit. The FBCP module also provides the means for off-line operator-requested tuning so that the feedback control path may be updated to achieve optimal operation. As introduced in Sections 3.4 and 3.5, the FBCP consists of a feedback control path and a supervisory tuning system. The feedback control path implements an open multiloop control scheme based on fuzzy-PID controllers and a multivariable disturbance feedforward interaction compensator, both of which are scheduled to achieve wide-range operation. The supervisory tuning system adjusts the parameters of the fuzzy-PID controllers using a genetic algorithm, and calculates the elements of the interaction compensator, whenever the power-pressure operating policy of the unit changes and upon request by the operator.

Figure 6.1 Feedback control processor in the ICCS-MP.
To achieve openness in the design of the FBCP module, it was proposed for the feedback control path a structure that can be easily and systematically expanded or contracted, as required by the scope of the coordinated control system application at hand. In general, the feedback control structure is a PID-based multiloop scheme with a feedforward-equivalent loop interaction compensator between the controllers and the process. Control loops may be included or excluded and the compensator may be augmented or reduced as required. For the ICCS-MP case, there are three fuzzy-PID controllers and a 3-input-3-output compensator (Figure 6.2). The power, pressure, and level controllers provide the control signals for the fuel valve, $u_{1\text{fb}}$, steam valve, $u_{2\text{fb}}$, and feedwater valve, $u_{3\text{fb}}$, respectively. Then the compensator updates these control signals to produce the compensated feedback control signals $u_{1\text{fbc}}$, $u_{2\text{fbc}}$, and $u_{2\text{fbc}}$.

It should be kept in mind that the purpose of the compensator is not to produce a perfectly decoupled system, but to reduce the control loop interaction effects to a degree manageable by the controllers. This greatly simplifies the design of the compensator, which is built in such a way that the direct paths in the control loops are preserved, and interaction compensation factors are only added among control signals. The factor values are determined from steady-state input-output process data using the relative gain matrix technique; neither a dynamic mathematical model of the process, nor the solution of a
nonlinear optimization problem are required to design the compensator. Singular value analysis and simulation experiments show that the proposed compensation is numerically well-conditioned, effectively handles control loop interaction, and facilitates controller tuning.

The fuzzy-PID controllers merge the techniques of gain-scheduling and multi-mode control in a single mechanism, which retains the advantages of both techniques. For each controller, the amalgamation is carried out by a first-order Sugeno-type fuzzy system, whose inference mechanism takes care of parameter interpolation and controller switching simultaneously. In its basic configuration, the controller is scheduled using the power set-point as scheduling variable. The power operating range is partitioned taking into account the nonlinearity of the process dynamics. Partitions are accounted for using fuzzy sets which are represented by overlapped triangular membership functions.

Both the controllers and the interaction compensator are scheduled to achieve wide-range operation. A two-dimensional scheduling strategy along and across the power-pressure operating window of the power unit is proposed to harmonize the FBCP with the set-point based process optimization strategy of the ICCS-MP, which depends on the current operating policy to be satisfied. Numerical analysis and simulation experiments demonstrate the feasibility of the proposed approach to achieve smooth wide-range operation. In what follows attention is first paid to the design of the feedforward compensation, then to the fuzzy-PID controllers, and their tuning strategy, and finally, to the scheduling strategy.

### 6.2 Decoupled Multiloop Control and Compensator Design

It was shown in Section 3.4.4 that the disturbance feedforward compensation functions, in the case of a multivariable controller consisting of multiple independent loops, can be arranged as an interaction compensator placed between the controllers and the process. This approach may then be equivalent to that of decoupling control schemes,
where the process dynamics are artificially augmented to appear as a set of independent processes that can be controlled independently. Nevertheless, the objective of the proposed compensation strategy is not to achieve perfect decoupling, but only to reduce the interaction effects among the control loops to alleviate the control effort demanded on the controllers. In addition, it is of great importance for the objectives of the overall power unit control strategy to implement a compensator with an open structure, also valid for wide-range interaction compensation. To this aim, a few necessary remarks on decoupling control structures are presented to sustain the design of the interaction compensator.

6.2.1 Decoupling control

Decoupling control of linear multivariable processes considers three main approaches: ideal decoupling, simplified decoupling, and inverted decoupling [Luyben 1970]. To describe these approaches in a simple way, a two-input-two-output process is used in what follows.

The input-output process relationships are given by:

\[
\begin{align*}
y_1 &= P_{11}u_1 + P_{12}u_2 \\
y_2 &= P_{21}u_1 + P_{22}u_2
\end{align*}
\]  

(6.1a, 6.1b)

where \(y_1\) and \(y_2\) are the process outputs, \(u_1\) and \(u_2\) are the process inputs, and \(P_{11}, P_{12}, P_{21},\) and \(P_{22}\) are the input-output process transfer functions.

The ideal decoupling structure is shown in Figure 6.3. The process inputs, \(u_1\) and \(u_2\), are related to the controller output signals, \(v_1\) and \(v_2\), through:

\[
\begin{align*}
u_1 &= D_{11}v_1 + D_{12}v_2 \\
u_2 &= D_{21}v_1 + D_{22}v_2
\end{align*}
\]  

(6.2a, 6.2b)

where \(D_{11}, D_{12}, D_{21},\) and \(D_{22}\) are the compensator transfer functions to be designed.
The input-output representation of the apparent plant (process and compensator) is obtained by substituting (6.2) into (6.1):

\[
y_1 = (P_{11}D_{11} + P_{12}D_{21})v_1 + (P_{11}D_{12} + P_{12}D_{22})v_2 \\
y_2 = (P_{21}D_{11} + P_{22}D_{21})v_1 + (P_{21}D_{12} + P_{22}D_{22})v_2
\]

(6.3a)

(6.3b)

which are required to appear as two independent systems, usually of the form:

\[
y_1 = P_{11}v_1 \\
y_2 = P_{22}v_2
\]

(6.4a)

(6.4b)

Then, equating coefficients in (6.3) and (6.4), and solving for the compensator transfer functions yields:

\[
D_{11} = \frac{P_{11}P_{22}}{P_{11}P_{22} - P_{12}P_{21}} \\
D_{12} = -\frac{P_{12}P_{22}}{P_{11}P_{22} - P_{12}P_{21}}
\]

(6.5a)

(6.5b)
where it should be highlighted that in the ideal decoupling case the compensator transfer functions (6.5) are more complicated than the apparent process transfer functions (6.4).

The simplified decoupling structure is shown in Figure 6.4. Compared to the ideal decoupling case, $D_{11}$ and $D_{22}$ in (6.2) are set equal to unity, and the coefficients of the crossed terms in (6.3) are set to zero. Taking these actions, $D_{12}$ and $D_{21}$ will be given by:

$$D_{12} = \frac{P_{12}}{P_{11}}$$  \hspace{1cm} (6.6a)  \\
$$D_{21} = \frac{P_{21}}{P_{22}}$$  \hspace{1cm} (6.6b)
In turn, the apparent transfer functions will be:

\[
\begin{align*}
y_1 &= \frac{P_{11}P_{22} - P_{12}P_{21}}{P_{22}} v_1 \\
y_2 &= \frac{P_{11}P_{22} - P_{12}P_{21}}{P_{11}} v_2
\end{align*}
\]  
(6.7a)  
(6.7b)

where, compared to the ideal decoupling case, in the simplified decoupling case the compensator transfer functions are simpler to implement, but the apparent process transfer functions are more complicated.

The inverted decoupling structure is shown in Figure 6.5. In this case, the elements of the decoupler are defined as in simplified decoupling: \(D_{11} = D_{22} = 1\), and

\[
\begin{align*}
D_{12} &= -\frac{P_{12}}{P_{11}} \\
D_{21} &= -\frac{P_{21}}{P_{22}}
\end{align*}
\]  
(6.8a)  
(6.8b)
Thus the apparent process transfer functions became the same as in ideal decoupling:

\[ y_1 = P_{11}v_1 \]  
\[ y_2 = P_{22}v_2 \]

(6.9a)  
(6.9b)

Note that in the inverted decoupling case both the compensator transfer functions and the apparent process transfer functions have the simplest expressions.

As will be shortly shown, the interaction compensator proposed in this research takes the best of all three decoupling approaches. The design will be carried out as in simplified decoupling, implementation is carried out as in inverted decoupling, and the resultant apparent process transfer functions correspond to those of ideal decoupling.

6.2.2 Relative gain array

The Relative Gain Array (RGA), \( \Lambda \), is the most widely used method to quantify input-output interaction in multivariable systems [Bristol 1966]. For a system with \( n \) inputs and \( n \) outputs, the RGA is expressed in matrix-like form:

\[
\Lambda = \left[ \begin{array}{ccc}
\lambda_{ij} & \ldots & \lambda_{ij} \\
\vdots & \ddots & \vdots \\
\lambda_{ij} & \ldots & \lambda_{ij}
\end{array} \right] \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, n
\]

(6.10)

where \( i \) is the output index, and \( j \) is the input index, and each element of the RGA, \( \lambda_{ij} \), is a relative measure of interaction between the process output variable \( y_i \) and the process input variable \( u_j \), which is defined as the ratio of two steady-state gains:

\[
\lambda_{ij} = \frac{\left( \frac{\partial y_i}{\partial u_j} \right)_{\text{all loops open}}}{\left( \frac{\partial y_i}{\partial u_j} \right)_{\text{all loops closed except for } j \text{ loop}}}
\]

(6.11)

\[
= \left( \frac{\text{open-loop gain}}{\text{closed-loop gain}} \right)_{\text{for loop } i \text{ under control of } u_j}
\]
The RGA elements have several interesting properties [McAvoy 1983], including:

- The elements of the RGA sum up to 1 across any row, or down any column:
  \[
  \sum_{i=1}^{n} \lambda_{ij} = \sum_{j=1}^{n} \lambda_{ij} = 1
  \]  
  (6.12)

- The $\lambda_{ij}$ are dimensionless; therefore, neither units, nor the absolute values actually taken by the variables $u_j$, or $y_i$, affect it.

- The value of $\lambda_{ij}$ is a measure of the steady-state interaction expected in the $i$-th loop of the multivariable system if its output $y_i$ is paired with $u_j$. In particular, $\lambda_{ij} = 1$ implies that $u_j$ affects $y_i$ without interaction with other control loops. Similarly, $\lambda_{ij} = 0$ implies that $u_j$ has absolutely no effect on $y_i$.

- Since the relative gains tell how much the gain of one loop changes when the other loops are closed, then they provide an indication on how the behavior of the loop will be changed because of the interactions. That is, the loop gain with all other loops closed, $K_{Cij}$, is given in terms of the open loop gain, $K_{Oij}$, by:
  \[
  K_{Cij} = \left(1/\lambda_{ij}\right)K_{Oij}
  \]  
  (6.13)

Several methods have been developed to calculate the relative gains. The gains can be determined either from experimental data or using a mathematical model of the process. The fact that a dynamic mathematical model of the process is not strictly necessary to calculate the relative gains makes it very attractive for practical applications. Nevertheless, and due to the basic scope of this dissertation, a mathematical approach will be used. The experimental approach will be explained for a 2x2 process only for completeness of this presentation. After that, the mathematical approach is described.

The experimental method uses input-output steady-state process data ($u_1$, $u_2$, $y_1$, $y_2$) about an operating point ($u_{1o}$, $u_{2o}$, $y_{1o}$, $y_{2o}$) to build an incremental steady-state model of the process:

\[
\tilde{y}_1 = K_{11}\tilde{u}_1 + K_{12}\tilde{u}_2 \\
\tilde{y}_2 = K_{21}\tilde{u}_1 + K_{22}\tilde{u}_2
\]  
(6.14a)

\[
\tilde{y}_1 = K_{11}\tilde{u}_1 + K_{12}\tilde{u}_2 \\
\tilde{y}_2 = K_{21}\tilde{u}_1 + K_{22}\tilde{u}_2
\]  
(6.14b)
where, for $i=1,2$, and $j=1,2$:

\[ \tilde{y}_i = y_i - y_{io} \quad (6.15a) \]
\[ \tilde{u}_i = u_i - u_{io} \quad (6.15b) \]
\[ K_{ij} = \frac{y_i - y_{io}}{u_j - u_{jo}} \quad (6.15c) \]

Then, the relative gains can be found from the definition in (6.11). First, to calculate $\lambda_{11}$ note that from (6.14a):

\[ \left( \frac{\partial y_1}{\partial u_1} \right)_{\text{all loops open}} = K_{11} \quad (6.16) \]

and ‘loop 2 closed in steady state’ implies the incremental output is zero, thus (6.14b) can be solved for $u_2$ in terms of $u_1$, and substituted back in (6.14a) to yield:

\[ \left( \frac{\partial y_1}{\partial u_1} \right)_{\text{loop 2 closed}} = K_{11} \left( 1 - \frac{K_{11}K_{21}}{K_{11}K_{22}} \right) \quad (6.17) \]

Hence from (6.11), (6.16) and (6.17):

\[ \lambda_{11} = \frac{1}{1 - \xi} \quad (6.18) \]

where:

\[ \xi = \frac{K_{12}K_{21}}{K_{11}K_{22}} \quad (6.19) \]

Second, the other relative gains can be found using the property that any column or row of the RGA adds up to 1. Therefore the RGA is given by:

\[ \Lambda = \begin{bmatrix} \lambda & 1 - \lambda \\ 1 - \lambda & \lambda \end{bmatrix} \quad (6.20) \]

where $\lambda = \lambda_{11}$.

The mathematical method [Ogunnaike and Ray 1994] uses the process transfer matrix model, $G(s)$. The procedure is shown here assuming an open-loop stable dynamic
and causal (physical) process. Later, when applied to the FFPU, it will be shown how to proceed in the open-loop unstable case. First, calculate the process gain matrix, $K$:

$$K = \lim_{s \to 0} G(s)$$

(6.21)

with elements $K_{ij}$. Then, assuming $K$ is invertible, calculate the transpose of the inverse of the gain matrix, $R$:

$$R = (K^{-1})^T$$

(6.22)

with elements $R_{ij}$. Finally, the RGA is obtained through the Hadamard product (element by element product) of matrices $K$ and $R$:

$$\Lambda = [\lambda_{ij}] = [K_{ij} R_{ij}]$$

(6.23)

### 6.2.3 Compensator design

In this section it is shown how the interaction compensator is designed for the ICCS-MP at a single operating point. The application range of the compensator will be extended to the whole power unit operating window in Section 6.4.2. Also, note that the compensator for the ICCS-MP is a $3 \times 3$ system, but the design method is general and can be applied straightforward to $n \times n$ systems, as required for an open system.

The interaction compensator is designed as a static decoupling compensator, that is, instead of using the process dynamic transfer functions, only the steady-state gain of the process transfer functions are used. The main advantages of static, or steady-state, decoupling are that the design involves simple numerical computations, the resulting compensators are always realizable, and the design only requires knowledge of the process steady-state gain matrix, which may be calculated from experimental steady-state input-output process data.

The interaction compensator takes into account the best characteristics of the three decoupling approaches presented in Section 6.2.1. The compensator is designed as
in the simplified decoupling case, and is implemented with the inverse decoupling structure to approximate, in steady-state, the simple fully decoupled apparent transfer functions of the ideal decoupling case.

Given the FFPU transfer matrix model, $T(s)$, as defined in (2.55): \[
T(s) = \begin{bmatrix}
T_{11}(s) & T_{12}(s) & T_{13}(s) \\
T_{21}(s) & T_{22}(s) & T_{23}(s) \\
T_{31}(s) & T_{32}(s) & T_{33}(s)
\end{bmatrix}
\]
(6.24)

where the elements $T_{ij}(s)$, for $i=1, 2, 3$, and $j=1, 2, 3$, were defined by (2.56a) through (2.56i). The steady-state gain matrix is given by:

\[ K = \lim_{s \to 0} T(s) \]
(6.25)

whose elements are calculated using (2.56a) through (2.56i) as:

\[ K_{11} = \lim_{s \to 0} T_{11}(s) = \frac{A_{12}B_{21}}{A_{11}A_{22}} \]
(6.26a)

\[ K_{12} = \lim_{s \to 0} T_{12}(s) = \frac{A_{12}B_{22} - A_{22}B_{12}}{A_{11}A_{22}} \]
(6.26b)

\[ K_{13} = \lim_{s \to 0} T_{13}(s) = \frac{A_{12}B_{23}}{A_{11}A_{22}} \]
(6.26c)

\[ K_{21} = \lim_{s \to 0} T_{21}(s) = -\frac{B_{21}}{A_{22}} \]
(6.26d)

\[ K_{22} = \lim_{s \to 0} T_{22}(s) = -\frac{B_{22}}{A_{22}} \]
(6.26e)

\[ K_{23} = \lim_{s \to 0} T_{23}(s) = -\frac{B_{23}}{A_{22}} \]
(6.26f)

\[ K_{31} = \lim_{s \to 0} T_{31}(s) = \lim_{s \to 0} \left( \frac{A_{32}B_{21}C_{33}}{sA_{22}} \right) = \lim_{s \to 0} \frac{Y_{31}}{s} \]
(6.26g)

\[ K_{32} = \lim_{s \to 0} T_{32}(s) = \lim_{s \to 0} \left( \frac{A_{22}B_{32}C_{33} - A_{13}B_{22}C_{33}}{sA_{22}} \right) = \lim_{s \to 0} \frac{Y_{32}}{s} \]
(6.26h)
where $\gamma_1$, $\gamma_2$, and $\gamma_3$, are appropriately defined for the equalities to hold.

From (6.26g)-(6.26i), it can be seen that the steady-state matrix is undetermined. This problem is always present when level dynamics (integrative processes) are involved (i.e., drum water level, dearator water level, and condenser water level in a power plant). Fortunately, the RGA provides a simple mechanism to deal with these cases, and allows the definition of an apparent gain matrix that can be used to design the interaction compensator. This fact has the very important implication that the control loops for natural integrative processes can be included in the overall design without further complication, thus adding for the generality of the approach. It should be emphasized that this kind of processes are normally avoided by other control strategies (i.e., predictive control), thus making them less desirable for the development of open generalized overall control strategies.

To calculate the RGA, the gain matrix, $K$, may be written as:

$$K = \lim_{s \to 0} \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix} = \lim_{s \to 0} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{s} \end{bmatrix} \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$

(6.27)

then,

$$K^{-1} = \lim_{s \to 0} \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix}$$

(6.28)

Introducing $L$, appropriately defined:

$$K^{-1} = \lim_{s \to 0} \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & s \end{bmatrix} = \lim_{s \to 0} \begin{bmatrix} L_{11} & L_{12} & L_{13}s \\ L_{21} & L_{22} & L_{23}s \\ L_{31} & L_{32} & L_{33}s \end{bmatrix}$$

(6.29)

then,
\[
\left( K^{-1} \right)^T = \lim_{s \to 0} \begin{bmatrix}
L_{11} & L_{21} & L_{31} \\
L_{12} & L_{22} & L_{32} \\
L_{13} & L_{23} & L_{33} \\
\end{bmatrix}
\quad (6.30)
\]

The RGA is obtained from (6.30) and (6.27) through the Hadamard product, canceling the \(s\), and taking limit, as:

\[
\Lambda = \left( K^{-1} \right)^T \ast K = \begin{bmatrix}
L_{11}K_{11} & L_{21}K_{12} & L_{31}K_{13} \\
L_{12}K_{21} & L_{22}K_{22} & L_{32}K_{23} \\
L_{13}^* & L_{23}^* & L_{33}^*
\end{bmatrix}
\quad (6.31)
\]

which would be the same if \(K\) was simply given by:

\[
K = \begin{bmatrix}
K_{11} & K_{12} & K_{13} \\
K_{21} & K_{22} & K_{23} \\
\gamma_{31} & \gamma_{32} & \gamma_{33}
\end{bmatrix}
\quad (6.32)
\]

Hence, (6.32) can be taken as the gain matrix of the FFPU to design the interaction compensator.

Next, from Figure 6.4, it can be seen that the simplified decoupling approach, as would be applied in steady-state, requires the product of the FFPU gain matrix and the interaction compensator to be of the form:

\[
\begin{bmatrix} M_1 & 0 & 0 \\
0 & M_2 & 0 \\
0 & 0 & M_3 \end{bmatrix} \ast \begin{bmatrix}
K_{11} & K_{12} & K_{13} \\
K_{21} & K_{22} & K_{23} \\
\gamma_{31} & \gamma_{32} & \gamma_{33}
\end{bmatrix} \begin{bmatrix} 1 & D_{12} & D_{13} \\
D_{21} & 1 & D_{23} \\
D_{31} & D_{32} & 1 \end{bmatrix} = 0
\quad (6.33)
\]

where \(M_1, M_2, \) and \(M_3\), would be the decoupled steady-state gains for the power, pressure, and level control loops, respectively. The \(D_{ij}, i=1,2,3, j=1,2,3,\) are the decoupling factors, or interaction compensation factors, to be determined.

Carrying out the product on the right side of (6.33), and equalizing to zero the resulting off-diagonal elements, yields the following equation systems:

\[
0 = K_{21} + K_{22}D_{21} + K_{23}D_{31} \quad (6.34a)
\]

\[
0 = K_{31} + K_{32}D_{21} + K_{33}D_{31} \quad (6.34b)
\]
\[ 0 = K_{11}D_{12} + K_{12} + K_{13}D_{32} \]  
\[ (6.35a) \]

\[ 0 = K_{31}D_{12} + K_{32} + K_{33}D_{32} \]  
\[ (6.35b) \]

\[ 0 = K_{11}D_{13} + K_{12}D_{23} + K_{13} \]  
\[ (6.36a) \]

\[ 0 = K_{21}D_{13} + K_{22}D_{23} + K_{23} \]  
\[ (6.36b) \]

where \( K_{31} = \gamma_{31}, K_{32} = \gamma_{32}, \) and \( K_{33} = \gamma_{33}. \) The equation systems can be solved simultaneously, by pairs, giving:

\[ D_{21} = \frac{K_{23}K_{31} - K_{21}K_{33}}{K_{22}K_{33} - K_{23}K_{32}} \]  
\[ (6.37a) \]

\[ D_{31} = \frac{K_{21}K_{32} - K_{22}K_{31}}{K_{22}K_{33} - K_{23}K_{32}} \]  
\[ (6.37b) \]

\[ D_{12} = \frac{K_{13}K_{32} - K_{12}K_{33}}{K_{11}K_{33} - K_{13}K_{31}} \]  
\[ (6.37c) \]

\[ D_{32} = \frac{K_{12}K_{31} - K_{11}K_{32}}{K_{11}K_{33} - K_{13}K_{31}} \]  
\[ (6.37d) \]

\[ D_{13} = \frac{K_{12}K_{23} - K_{13}K_{22}}{K_{11}K_{22} - K_{12}K_{21}} \]  
\[ (6.37e) \]

\[ D_{23} = \frac{K_{13}K_{21} - K_{11}K_{23}}{K_{11}K_{22} - K_{12}K_{21}} \]  
\[ (6.37f) \]

Relations (6.37a) through (6.37f) define the desired interaction compensator in (6.33), whose implementation is carried out as in the inverse decoupling approach (Figure 6.6). Hence, as for the ideal decoupling case, the steady-state gains of the decoupled FFPU will be:

\[ M_1 = K_{11} \]  
\[ (6.38a) \]

\[ M_2 = K_{22} \]  
\[ (6.38b) \]

\[ M_3 = K_{33} \]  
\[ (6.38c) \]
6.2.4 Decoupling feasibility and sensitivity

The feasibility of the interaction compensator can be assessed by analyzing the numerical characteristics of the apparent FFPU steady-state gain matrix. To that aim, designing the compensator, $D$, is equivalent to solve the system of linear equations:

$$DvMvKu = -1$$  \hspace{1cm} (6.40)

Clearly, if $K$ is degenerate, decoupling will be extremely difficult to achieve, and if $K$ is not full-rank decoupling can not be achieved. Degeneracy of matrix $K$ can be

Figure 6.6 Inverse decoupling implementation of interaction compensator.
assessed through the condition number, which is the single most reliable indicator of the conditioning of a matrix. The condition number, $cn$, is defined as the ratio of the largest to the smallest singular value of the gain matrix:

$$cn = \frac{\sigma_{\max}(K)}{\sigma_{\min}(K)}$$ (6.41)

The larger the condition number, the poorer the numerical conditioning of the matrix, that is, the larger its degeneracy. Plotted in Figure 6.7 are the condition numbers of the process gain matrix for the operating points of the constant and lowest pressure operating policies in Tables 2.6 and 2.9, respectively. Clearly, with those values the gain matrix does not degenerate; thus a compensator can be securely designed at any operating point on and between those policies, which by the way constitutes the practical operating zone of FFPUs. Less favorable condition numbers might be obtained at higher pressures.

![Figure 6.7 Gain matrix condition number window.](image)

After the feasibility of the interaction compensator has been established for the FFPU, another closely related issue to examine is the sensitivity of the compensator to modeling errors, since no process model is 100% accurate. Considering an error $\Delta K$ in the estimate of the steady-state gain matrix:
the compensator $D$, designed in terms of $K$, yields:

$$y = (K + \Delta K) Dv = (K + \Delta K) K^{-1} Mv$$ \hspace{1cm} (6.43)

where the amount of error introduced into $y$ due to the model mismatch is given by:

$$\Delta y = \Delta KK^{-1} Mv$$ \hspace{1cm} (6.44)

that can be rewritten using the definition of the inverse of a matrix as:

$$\Delta y = \frac{\Delta K Adj(K) Mv}{|K|}$$ \hspace{1cm} (6.45)

Clearly, if the determinant of the gain matrix, $|K|$, is very small, small modeling errors will be magnified into very large errors in $y$, and small changes in the controller output will also result in large errors in $y$. Plotted in Figure 6.8 are the determinant of the gain matrix, $|K|$, at the operating points of the constant and lowest pressure operating policies in Tables 2.6 and 2.9, respectively. From these values it can be seen that the sensitivity to modeling errors is reasonable in the practical operating zone of the FFPU. Hence, compensators may be confidently designed at any operating point in there.

![Figure 6.8 Gain matrix determinant value window.](image-url)
6.3 Scheduled Fuzzy-PID Controller

6.3.1 Gain scheduling and multimode control

Gain scheduling has been pointed out to be an effective strategy for controlling nonlinear processes over ample operating regions when the process dynamics and the operation maneuvers are reasonably well-known in advance [Astrom and Wittenmark 1989]. Normally, in this approach a linear controller is designed for the linear model of the process at several selected operating points. Then, during operation the parameters of the controller are updated as the operating conditions change along the selected operating points (Figure 6.9).

![Figure 6.9 Gain scheduling control.](image)

Although there is no generally accepted design procedure, the design of a gain-scheduled controller typically proceeds in several steps. First, a variable \( \alpha \), well correlated with the changes in the process dynamics, should be identified to be used as the scheduling variable. Preferably, this variable should be readily available and its time dependency should be easily handled. Second, a set of operating points should be chosen to span the whole operating range of the process. This set defines an array of values...
A\{\alpha_1,\ldots,\alpha_n\} in the scheduling variable and a partition of the operating range. Third, design the linear controller at each operating point, using the linear time-invariant models of the plant at the selected points if required, and store the controller parameters. Fourth, design the gain-scheduling scheme to interpolate the controller parameters for operation between the selected operating points.

The main advantages of gain scheduling are that the plethora of linear control methods can be used in the design of the control system, and that the controller parameters can be updated very rapidly, since no parameter estimation is needed. Major difficulties might include the selection of the scheduling variable, optimal partition of the operating space, large amount of time required to synthesize the controllers for cases with large number of partitions, and the selection of the parameter scheduling procedure. The former problem may result in unsatisfactory or even unstable performance, which can be accentuated by the interpolation strategy, sharpness of the partition boundaries, and non-slowly varying process dynamics.

Despite the many practical implementations of gain scheduled controllers, there has always been a major concern about the lack of theoretical results about the global properties of the controlled system, that is, although the system could have excellent robust stability and performance properties at the selected operating points, there is no guarantee that these properties will hold at all points between the selected points. Nevertheless, it should be noted that even theoretically well-supported control methods are of a local nature once a real application is attempted. Thus in general, and not exclusively for gain scheduling, the control system properties for wide-range operation are demonstrated through extensive simulation experiments, which may become an overwhelming task when many points of interest are considered.

In the above regard, [Rough 1991] provides analytical results showing that in the typical gain-scheduling scheme, where an internal process variable is used to determine the process operating condition, a slowly-varying scheduling variable guarantees that the process state and output do not deviate far away from the values at the selected operating points. Additionally, in [Shamma and Athans 1990] it is shown that scheduling on a
reference trajectory and on the plant output maintains global properties provided that the scheduling variables have a sufficiently slow time variation, the unmodeled nonlinearities are small, and the unmodeled dynamics are not excited significantly. As will be later shown through simulation experiments, all these conditions are met at a FFPU during wide-range load tracking in the normal operating mode.

Another attractive approach for wide-range control of nonlinear processes is multimode control [Yamaguchi and Hirai 1998]. In this approach, switching among several controllers, each designed for a partition of the operating space, is made in accordance to the current operating conditions (Figure 6.10).

![Multi-mode control diagram]

The multimode approach provides great versatility on the control techniques that can be used in each controller, and from a wider perspective, also allows the realization of reconfigurable schemes that can even have different process input-output pairing. This versatility could be a great advantage for the implementation of control strategies across different operating states (i.e., normal, emergency, trip, restoration, etc.) and operating stages (cold startup, hot standby, loading, shutdown, etc.) within the same operating state. Very possibly, the major drawback might be that of implementing the switching logic to
achieve smooth transfer between two different controllers and to avoid intermittent operation of two controllers in adjacent operating regions with sharp boundaries. It should be noted that the multimode approach may be equivalent to gain scheduling provided that the controllers have the same structure.

In both control methods described above, the implementation of the interpolation strategy and the switching logic are key issues for the success of the control scheme to achieve satisfactory wide-range operation. As an alternative solution to these problems in PID-based industrial process control systems, this section introduces a fuzzy PID-like controller that merges the techniques of gain-scheduling and multimode control in a single mechanism, retaining the advantages of both techniques. The amalgamation is carried out by a TSK-type fuzzy system (Section 5.2.1), whose inference mechanism takes care of parameter scheduling and controller switching simultaneously. The fuzzy-PID controller, as will be called hereafter, can directly substitute the usual PID controllers in an already existing control configuration. Hence, the fuzzy-PID controller will be used to extend the operating range of the multiloop control system of a FFPU.

6.3.2 Fuzzy PID controller

Essentially, the proposed fuzzy-PID controller is a fuzzy system that implements a multimode controller, where all the component controllers are of the usual PID kind, and each PID controller is associated to a partition of the FFPU’s operating space. A noteworthy feature of the fuzzy-PID controller is that the switching logic, based on the fuzzification and inference mechanisms of a TSK-type fuzzy system (Section 5.2.1), is equivalent to the parameter interpolation effect driven by the scheduling variable in a gain-scheduling controller. The structure and operation of the fuzzy-PID controller is described in what follows.

The core of the fuzzy-PID controller is a four-input-one-output TSK-type fuzzy system with inference rules of the form:
where \( r \) is the rule number, \( x_1, x_2, x_3, \) and \( x_4 \), are the system inputs, \( L_X \) is a linguistic term of the input \( x_1 \), and \( c_2, c_3 \) and \( c_4 \) are constants to be defined to calculate the rule output \( u \). Note that only the first input \( x_1 \) is fuzzyfied and its linguistic terms \( L_X \) define overlapping partitions on the universe of discourse of \( x_1 \). Therefore, to implement the multimode controller, \( x_1 \) is used as the scheduling variable \( \alpha \) in a gain scheduling control, and the linguistic terms \( L_X \) are a very convenient way to represent the partitions, \( A \), of the unit’s operating space, as overlapping fuzzy sets along the range of the scheduling variable. Additionally, the constants \( c_2, c_3 \) and \( c_4 \) are used as the PID controller parameters \( K_p, K_i, \) and \( K_d \), representing the proportional, integral, and derivative gains, respectively. The inputs \( x_2, x_3, \) and \( x_4 \), are associated to the driving error signal, the error integral, and the error derivative:

\[
\begin{align*}
x_2 & \triangleq e = r - y & (6.47a) \\
x_3 & \triangleq \int e dt & (6.47b) \\
x_4 & \triangleq \frac{de}{dt} & (6.47c)
\end{align*}
\]

Thus each component controller is represented by a rule of the form:

\[
\text{IF } \alpha \text{ is } A_r, \text{ THEN } u_r = K_p e + K_i \int e dt + K_d \frac{de}{dt} \quad (6.48)
\]

where the scheduling variable \( \alpha \), to be selected in accordance with the application of the controller, represents an operating point, and the linguistic term \( A_r \) is a fuzzy set that represents a partition of the unit’s operating space along the space of the scheduling variable. Normally, a partition will correspond to a physical operating condition that has no crisp boundaries with its neighbor operating conditions. This corresponds to the fuzzy set membership functions showing overlap with the adjacent membership functions.

As for any regular TSK-type fuzzy system, the control signal provided by the fuzzy-PID controller is the weighted average of the control signals calculated by each rule:
\[ u = \frac{\sum_{r} w_r u_r}{\sum_{r} w_r} \]  
(6.49)

where the rule weights \( w_r \) are calculated as the product of the input membership values for each rule:

\[ w_r = \prod_{i=1}^{n} \mu_{X_i}(x_i) \]  
(6.50)

where \( i=1, \ldots, n \) is the input number. Nevertheless, since in the fuzzy-PID controller case there is only one input variable being fuzzyfied, the product in (6.50) has only one factor:

\[ w_r = \mu_{\alpha}(\alpha) \]  
(6.51)

where \( \alpha \) is the scheduling variable (first input) that takes a crisp value at the current operating point. Furthermore, assuming normal membership functions, \( \max(\mu(\alpha))=1 \), with triangular shape, at most two fuzzy sets contain any given operating point \( \alpha_0 \) simultaneously, as shown in Figure 6.11 This situation means that at any given operating point only two rules are fired, or equivalently two controllers are active, at most, and the sums in (6.49) have two terms at most. Assuming the controllers \( k \) and \( k-1 \), associated with the \( k \) and \( k-1 \) operating regions (membership functions), are active:

\[ \mu_{\alpha_0}(\alpha_0) + \mu_{\alpha}(\alpha_0) = 1 \]  
(6.52)

Then, the denominator in (6.49) is unity, and the controller output at \( \alpha_0 \) is given by:

\[ u(\alpha_0) = \mu_{\alpha_0}(\alpha_0) u_{k-1} + \mu_{\alpha}(\alpha_0) u_k \]  
(6.53)

where each rule output is given in terms of controller inputs as:

\[ u_{k-1} = K_p(k-1)e + K_i(k-1) \int edt + K_d(k-1) \frac{de}{dt} \]  
(6.54a)

\[ u_k = K_p(k)e + K_i(k) \int edt + K_d(k) \frac{de}{dt} \]  
(6.54b)

Therefore, the output of the controller (6.53) can be written as:
where the equivalent controller parameters are scheduled as:

\[ K_{pe} = (1 - \mu_k) K_{p(k-1)} + \mu_k K_{p(k)} \]  
\[ K_{ie} = (1 - \mu_k) K_{i(k-1)} + \mu_k K_{i(k)} \]  
\[ K_{de} = (1 - \mu_k) K_{d(k-1)} + \mu_k K_{d(k)} \]

where \( \mu_k \) is the membership function value of the scheduling variable in the fuzzy set \( A_k \) corresponding to the partition \( k \).

The resulting PID control law in (6.55) can be made more versatile using the following reference-weighted driving errors:

\[ e_p = w_p r - y \]  
\[ e_d = w_d r - y \]

in the proportional and derivative terms, respectively. The integral term keeps using the true error to eliminate steady-state errors. The weight values \( w_p \) and \( w_d \) affect the response to changes in the reference value. The weight \( w_p \) may be adjusted toward zero to reduce overshoot initiated by large changes in the reference value. The weight \( w_d \) is

![Figure 6.11 Rule firing for any given operating condition.](image-url)
Normally set to zero to avoid large error rate of change due to sudden changes in the reference value. In any case, the response to load disturbances and measurement noise remains the same as with the true error. Large step changes in the reference value are generally not allowed during normal operation in industrial applications, thus setting \( w_p = 1 \) and \( w_d = 0 \), the final control output provided by the fuzzy-PID controller is:

\[
u = K_p e + K_i \int e dt - K_d \frac{dy}{dt}
\]

and the inference rules take the form:

\[
IF \ \alpha \ is \ \mathbb{A}_r, \ THEN \ u_r = K_p e + K_i \int e dt - K_d \frac{dy}{dt}
\]

The results of this section are summarized in the diagram of the proposed fuzzy-PID controller in Figure 6.12, where it should be emphasized that the scheduling variable in the antecedent of each rule implements a control switching mechanism that determines the controllers to be active, while the composition of the final output control signal is mathematically equivalent to a gain scheduling mechanism with interpolation of parameter values.

Figure 6.12  Fuzzy-PID controller.
6.3.3 Application of the fuzzy-PID controller

With respect to the design procedure of a gain-scheduling controller summarized in Section 6.3.1, the steps regarding the design of the linear controller and the design of the gain-scheduling scheme (including the controller switching logic) were just covered in Section 6.3.2. This section includes the steps regarding the selection of the scheduling variable and the operating points of interest, which are specific to the application of the fuzzy-PID controller to the FFPU. Because of its nature, the controller tuning phase, which requires all the components of the ICCS-MP already in place, is postponed until Chapter 7 where they all get integrated.

The first relevant issue for the application of the fuzzy-PID controller to the FFPU is that a reference signal, the power set-point $E_d$, can be used as the scheduling variable, unlike typical gain-scheduling schemes that use other process variables. It should be noted that the power reference is readily available, it is a slowly-varying variable whose time dependency is easily handled, and it is well correlated to the changes in the process dynamics, thus making it the perfect candidate to be the scheduling variable $\alpha$. To support the previous statements note that:

- The power reference $E_d$ is always made equal to the unit load demand $E_{uld}$, which is the signal used to command the FFPU from upper hierarchical levels, and whose values and time dependence are always neatly specified and known.
- Most control systems at FFPUs already have provisions to independently set both the power set-point and the load rate, and the generated power and its rate of change are always very closely monitored as a routine procedure.
- At FFPUs only ramps, with a maximum slope of 5% of the base load per minute, are used to change the power set-point; for practical and safety reasons no step changes are allowed.
- It is commonly observed that during normal operation there are no large output power deviations from the commanded reference signal; this meaning that the process dynamics are not far from the values corresponding to the desired output power.
• The power reference is well correlated to the process dynamics since, as was shown in Chapter 2, most of the non-zero elements in the linear model system matrix, $A$, depend on the actual main steam pressure value, $x_2$, at the given operating point, which is regulated at a value specified by the pressure set-point $P_{d}$, which in a coordinated control scheme is obtained through a predefined fixed mapping from the unit load demand, $E_{uld}$, which turns out to be equal to the power reference $E_{d}$.

Consequently, using the power set-point $E_{d}$ as a scheduling variable guarantees smooth trajectories in both the scheduled parameters and the process dynamics (process states), and stable process variable responses, as required for a successful application of the gain-scheduling strategy [Shamma and Athans 1990, Rough 1991].

Next step is to select the operating points of interest for scheduling. The approach to follow is to select the operating points after partitioning the operating space, for which it is contributed a systematic approach that takes into account the nonlinearity of the process. To this aim, it is assumed that the linear model, at the equilibrium points where it is calculated, is a good approximation to the process dynamics. Hence the eigenvalues of the linear model system matrix (2.53a, repeated here for convenience):

$$A = \begin{bmatrix}
-0.1 & 1.125(0.073u_{2e} - 0.016)x_{2e}^{1/8} & 0 \\
0 & -0.002u_{2e}x_{2e}^{1/8} & 0 \\
0 & -0.012(1.1u_{2e} - 0.19) & 0
\end{bmatrix}$$

(6.60)

are taken as indicators of the process dynamics, and a sufficiently dense set of operating points along the whole operating range of the power unit is used to investigate the behavior of the system eigenvalues for any given power-pressure operating policy. The eigenvalue with the largest magnitude variations is considered to be representative of the system nonlinearity, and so, will be used to define the partitions of the operating range. Figure 6.13 plots such eigenvalue as a function of the power reference for the typical sliding-pressure operating policy defined in Table 2.7.

To proceed with the partition, the normalized magnitude of the largest-varying eigenvalue with respect to its value at base load conditions ($E=160$ MW, $P=140$ Kg/cm$^2$, $L=0$ cm) is defined as:
\[
neig(A) \triangleq \frac{|eig(A)|}{|eig(A_{\text{base-load}})|}
\]

where \( |eig(A)| \) is the magnitude of the largest-varying eigenvalue at the given operating conditions, and \( |eig(A_{\text{base-load}})| \) is the magnitude of the largest-varying eigenvalue at base-load conditions. The normalized eigenvalue is plotted as a function of power in Figure 6.14.

Then, the operating range partitions can be defined in terms of a given percentage variation along the eigenvalue range. For instance, dividing the eigenvalue variation range in 5 equal-size sections (20 % of the whole eigenvalue variation range each) will divide the power operating range in 5 sections by projection through the nonlinear relation between the scheduling variable and the normalized eigenvalue magnitudes as shown also in Figure 6.14. Then, the operating points of interest are located at the middle point of the sections just found. Table 6.1 summarizes the values of the state-variables and process inputs at the selected operating points for this example. Finally, these points define the overlapping partitions of the operating space as fuzzy sets with the membership functions as shown in Figure 6.15. These fuzzy sets are to be used in the rule antecedents of the fuzzy-PID controllers.

After the selected operating points have been determined, and the fuzzy operating partitions have been defined, the parameters in the control rule consequents have to be determined, that is, the controllers should be tuned at the selected operating points. Treatment of this topic and the implementation of a supervisory tuning system is dealt with in Chapter 7.

<table>
<thead>
<tr>
<th>TABLE 6.1 SELECTED OPERATING POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>18.2160</td>
</tr>
<tr>
<td>36.8446</td>
</tr>
<tr>
<td>61.0997</td>
</tr>
<tr>
<td>94.1372</td>
</tr>
<tr>
<td>141.6661</td>
</tr>
</tbody>
</table>
Figure 6.13 Actual system eigenvalue with largest variation

Figure 6.14 Operating space partitioning.

Figure 6.15 Fuzzy sets of power scheduling variable.
6.4 Feedback Control Path Design

Operation of the FFPU at any point of the power-pressure operating window requires the design and implementation of a mechanism to schedule both the fuzzy-PID controllers and the interaction compensator in two dimensions (2D): power and pressure. In this regard, the scheduling strategy presented in Section 6.3.3 can be advantageously used with just a few minor considerations regarding the partition of the operating space and the selection of the operating points of interest for scheduling. In what follows a 2D scheduling mechanism for the fuzzy-PID controllers and the interaction compensator is described.

6.4.1 2D scheduling of fuzzy-PID controllers

To design the 2D scheduling mechanism to apply the fuzzy-PID controller in a multiloop control configuration at an FFPU, it should be bear in mind that the ICCS-MP always drives the power unit along a specific power-pressure operating policy, which can be changed across the power-pressure operating window to achieve optimal operation under different operating conditions. Thus, to guarantee consistency in partitioning the FFPU’s operating window and in the selection of the operating points for scheduling, along and across the whole operating window, a slightly different approach than that presented in Section 6.3.3 is followed: instead of defining first the operating partitions and then locate the selected operating points, the reverse order will be utilized, and as will be shortly shown, this approach will divide the operating space into zones having equivalent dynamics, thus requiring the same controller.

To start with, several power-pressure operating policies are taken into account to help in visualizing the 2D partitioning of the operating window. The cases of power-pressure operating policies being considered include the upper-pressure limit, constant pressure operating policy, sliding pressure operating policy, an arbitrary variable-pressure operating policy, and the lower-pressure limit, as shown in Figure 6.16. Note that the
power range under consideration comprehends from 10 MW, corresponding to minimum load during load operation mode, through 180 MW at maximum peak load.

As for the scheduling case along a single operating policy, the normalized value of the eigenvalue with the largest variation along the power operation range (6.61) is used as an index to account for the process nonlinearity. The actual eigenvalues and the normalized eigenvalues are plotted as functions of power for each power-pressure operating policy in Figures 6.17 and 6.18, respectively.

Partitioning of the operating window is carried out in terms of a variation percentage of the nonlinear dynamics of the plant as reflected through the normalized eigenvalues. For instance, if each partition of the operating window, along any power-pressure characteristic, is to account for 40% variation in the nonlinear dynamics of the plant, the partitions will be defined by partitioning the normalized eigenvalue range in 40% sections to span the total normalized eigenvalue range.
In addition, since the scheduling strategy requires overlapping partitions, each partition will overlap with its two adjacent neighbors, thus the partition centers will be allocated within 40%/2=20% separation in the normalized eigenvalue space. At the end, the centers of the partitions will be the selected operating points for scheduling. To locate the partitions, and their centers, a major operating point should be used as a reference. Base-load conditions constitute such a point located at the intersection of the constant pressure operating policy and the horizontal line of normalized eigenvalue with unit value. Then, continuing with the 40% partition example, the other operating points will be at the intersections of the horizontal lines with 20% separation, that is, at values 0.8, 0.6, and 0.4, as shown also in Figure 6.18. Note that the horizontal lines of constant normalized eigenvalue will define dynamically equivalent operating points at different power-pressure operating policies, assuming all other process dynamics (eigenvalues) remain constant. Next step is to map the points and partitions just determined onto the power operating space.
Taking each power-pressure policy individually, the operating points in the normalized eigenvalue space define operating points along the power operating space by projection through the normalized eigenvalue function, as shown in Figures 6.19, 6.22, 6.25, 6.28, and 6.31, for the different power-pressure characteristics being considered. Table 6.2 provides the numerical values of the operating points in the power axis. In turn, the power operating points define specific values of operating pressure through the corresponding power-pressure policies, as shown in Figures 6.20, 6.23, 6.26, 6.29, and 6.32. Table 6.2 also provides the numerical values of the pressure at the selected points. In summary, Table 6.2 provides the location of selected operating points along and across the power-pressure operating window.

The selected operating points are then used to define the fuzzy sets to represent the operating space partitions required by the fuzzy-PID controllers. For each operating policy, each internal power operating point, \( \alpha_k \), defines the center of a triangular membership function:

![Figure 6.18 Normalized eigenvalues along power-pressure characteristics.](image)
where \( E_d \) is the power set-point being used as scheduling variable by the fuzzy-PID controllers, and the two base points are defined by the adjacent power operating points: \( \alpha_{k-1} \) and \( \alpha_{k+1} \). Operating points at the ends (first and last operating point considered) define a left-open saturated ramp membership function for the first point:

\[
LE_1(E_d) = \begin{cases} 
1 & \text{if } E_d < \alpha_1 \\
\frac{\alpha_2 - E_d}{\alpha_2 - \alpha_1} & \text{if } \alpha_1 \leq E_d < \alpha_2 \\
0 & \text{if } \alpha_2 \leq E_d
\end{cases}
\] (6.63)

and a right-open saturated ramp membership function for the last, \( n \), point:

\[
LE_n(E_d) = \begin{cases} 
0 & \text{if } E_d < \alpha_{n-1} \\
\frac{E_d - \alpha_{n-1}}{\alpha_n - \alpha_{n-1}} & \text{if } \alpha_{n-1} \leq E_d < \alpha_n \\
1 & \text{if } \alpha_n \leq E_d
\end{cases}
\] (6.64)

Figures 6.21, 6.24, 6.27, 6.30, and 6.33 show the membership functions of the fuzzy sets corresponding to the partitions for each operating policy.

| TABLE 6.2 SELECTED POINTS FOR SCHEDULING (40% CASE) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| High pressure limit | Constant pressure | Sliding pressure | Variable pressure | Low pressure limit |
| E   | P   | E   | P   | E   | P   | E   | P   | E   | P   |
| 55.8 | 212.4 | 39.1 | 140.0 | 21.5 | 70.8 | 10.5 | 32.5 | 10.8 | 32.0 |
| 104.1 | 187.4 | 79.4 | 140.0 | 49.6 | 84.8 | 25.7 | 41.8 | 19.7 | 32.0 |
| 142.7 | 168.4 | 119.7 | 140.0 | 91.5 | 105.8 | 51.3 | 57.9 | 29.0 | 32.0 |
| 174.6 | 153.3 | 160.0 | 140.0 | 160.0 | 140.0 | 108.2 | 93.5 | 70.7 | 60.3 |
Figure 6.19  Partition of power range for upper-limit pressure.

Figure 6.20  Pressure operating points for upper-limit pressure.

Figure 6.21  Membership functions for upper-limit pressure.
Figure 6.22 Partition of power range for constant pressure.

Figure 6.23 Pressure operating points for constant pressure.

Figure 6.24 Membership functions for constant pressure.
Figure 6.25 Partition of power range for sliding pressure.

Figure 6.26 Pressure operating points for sliding pressure.

Figure 6.27 Membership functions for sliding pressure.
Figure 6.28 Partition of power range for variable pressure.

Figure 6.29 Pressure operating points for variable pressure.

Figure 6.30 Membership functions for variable pressure.
Figure 6.31 Partition of power range for lower-limit pressure.

Figure 6.32 Pressure operating points for lower-limit pressure.

Figure 6.33 Membership functions for lower-limit pressure.
The partitioning of the unit’s operating window can be summarized by the grid in Figure 6.34 that shows lines of points with equivalent dynamics, that is with constant normalized eigenvalues. The intersection points correspond to the selected operating points to be used for scheduling, and the lines correspond to the centers and base points (boundaries) of the overlapping partitions. Also, Figure 6.35 summarizes the membership functions for all five policy cases considered in the 40% partition example. It can be seen how the fuzzy sets evolve across the pressure values. This effect constitutes the scheduling strategy along the pressure axis. Note that while the scheduling along the power axis takes place on-line during the operation of the controller, the scheduling along the pressure axis will take place off-line any time the operating policy is changed. Because of this, it can be said that the 2D scheduling control strategy depends on the automation strategy of the power unit.

Figure 6.34  Partitioning of power-pressure operating window (40% case).
As another case study of the 2D partitioning and scheduling strategy, Figures 6.36 and 6.37 show the power-pressure operating window partitioning and scheduling of the fuzzy sets for the case with 20% variation in the unit dynamics. Table 6.3 summarizes the numerical values of the corresponding selected operating points.

**TABLE 6.3 SELECTED POINTS FOR SCHEDULING (20% CASE)**

<table>
<thead>
<tr>
<th>neig(A)</th>
<th>HP limit</th>
<th>Const. pressure</th>
<th>Sliding pressure</th>
<th>Variable pressure</th>
<th>LP limit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>E</td>
<td>P</td>
<td>E</td>
<td>P</td>
<td>E</td>
</tr>
<tr>
<td>0.3</td>
<td>26.5</td>
<td>228.2</td>
<td>18.9</td>
<td>140.0</td>
<td>10.6</td>
</tr>
<tr>
<td>0.4</td>
<td>55.8</td>
<td>212.4</td>
<td>39.1</td>
<td>140.0</td>
<td>21.5</td>
</tr>
<tr>
<td>0.5</td>
<td>81.5</td>
<td>199.0</td>
<td>59.2</td>
<td>140.0</td>
<td>34.4</td>
</tr>
<tr>
<td>0.6</td>
<td>104.1</td>
<td>187.4</td>
<td>79.4</td>
<td>140.0</td>
<td>49.6</td>
</tr>
<tr>
<td>0.7</td>
<td>124.4</td>
<td>177.3</td>
<td>99.5</td>
<td>140.0</td>
<td>68.4</td>
</tr>
<tr>
<td>0.8</td>
<td>142.7</td>
<td>168.4</td>
<td>119.7</td>
<td>140.0</td>
<td>91.5</td>
</tr>
<tr>
<td>0.9</td>
<td>159.4</td>
<td>160.4</td>
<td>139.8</td>
<td>140.0</td>
<td>121.1</td>
</tr>
<tr>
<td>1.0</td>
<td>174.6</td>
<td>153.3</td>
<td>160.0</td>
<td>140.0</td>
<td>160.0</td>
</tr>
</tbody>
</table>

Figure 6.35 Membership functions across the operating window (40% case).
Figure 6.36 Partitioning of power-pressure operating window (20% case).

Figure 6.37 Membership functions across the operating window (20% case).
6.4.2 2D scheduling of interaction compensator

The 2D scheduling of the interaction compensator elements is simpler than the fuzzy-PID controllers. The compensator elements shown in Figure 6.6 are calculated in terms of the power set-point $E_d$, the pressure set-point $P_d$, and the control signals $u_1$, $u_2$, and $u_3$, using relations (6.37a) through (6.37f), along the power-pressure operating policy being used. The compensator elements are recalculated off-line to fit new operating conditions any time the power-pressure operating policy is modified. Once calculated, each compensator element is implemented as a look-up table that provides the element value as a function of the power set-point as shown in Figure 6.38. Tables 6.4 through 6.8 provide sample values of the elements for the five power-pressure operating policies being considered in this chapter.

![Figure 6.38 Gain-scheduled interaction compensator.](image-url)
### TABLE 6.4 COMPENSATOR ELEMENTS FOR HIGH-PRESSURE LIMIT

<table>
<thead>
<tr>
<th>$E_d$</th>
<th>$P_d$</th>
<th>$D_{12}$</th>
<th>$D_{13}$</th>
<th>$D_{21}$</th>
<th>$D_{23}$</th>
<th>$D_{31}$</th>
<th>$D_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>237.39400</td>
<td>-7.48204</td>
<td>0.16667</td>
<td>0.80043</td>
<td>0.0</td>
<td>1.48240</td>
<td>-2.41971</td>
</tr>
<tr>
<td>20.0</td>
<td>231.83400</td>
<td>-3.53014</td>
<td>0.16667</td>
<td>0.82146</td>
<td>0.0</td>
<td>1.48571</td>
<td>-1.04731</td>
</tr>
<tr>
<td>30.0</td>
<td>226.33400</td>
<td>-2.21851</td>
<td>0.16667</td>
<td>0.84332</td>
<td>0.0</td>
<td>1.48907</td>
<td>-0.59459</td>
</tr>
<tr>
<td>40.0</td>
<td>220.89500</td>
<td>-1.56686</td>
<td>0.16667</td>
<td>0.86606</td>
<td>0.0</td>
<td>1.49248</td>
<td>-0.37174</td>
</tr>
<tr>
<td>50.0</td>
<td>215.51500</td>
<td>-1.17908</td>
<td>0.16667</td>
<td>0.88974</td>
<td>0.0</td>
<td>1.49593</td>
<td>-0.24076</td>
</tr>
<tr>
<td>60.0</td>
<td>210.19300</td>
<td>-0.92315</td>
<td>0.16667</td>
<td>0.91441</td>
<td>0.0</td>
<td>1.49945</td>
<td>-0.15567</td>
</tr>
<tr>
<td>70.0</td>
<td>204.93100</td>
<td>-0.74252</td>
<td>0.16667</td>
<td>0.94012</td>
<td>0.0</td>
<td>1.50301</td>
<td>-0.09677</td>
</tr>
<tr>
<td>80.0</td>
<td>199.72600</td>
<td>-0.60888</td>
<td>0.16667</td>
<td>0.96694</td>
<td>0.0</td>
<td>1.50664</td>
<td>-0.05421</td>
</tr>
<tr>
<td>90.0</td>
<td>194.57900</td>
<td>-0.50653</td>
<td>0.16667</td>
<td>0.99495</td>
<td>0.0</td>
<td>1.51032</td>
<td>-0.02251</td>
</tr>
<tr>
<td>100.0</td>
<td>189.49000</td>
<td>-0.42603</td>
<td>0.16667</td>
<td>1.02420</td>
<td>0.0</td>
<td>1.51407</td>
<td>0.00162</td>
</tr>
<tr>
<td>110.0</td>
<td>184.45800</td>
<td>-0.36140</td>
<td>0.16667</td>
<td>1.05479</td>
<td>0.0</td>
<td>1.51788</td>
<td>0.02026</td>
</tr>
<tr>
<td>120.0</td>
<td>179.48400</td>
<td>-0.30863</td>
<td>0.16667</td>
<td>1.08679</td>
<td>0.0</td>
<td>1.52176</td>
<td>0.03481</td>
</tr>
<tr>
<td>130.0</td>
<td>174.56900</td>
<td>-0.26496</td>
<td>0.16667</td>
<td>1.12029</td>
<td>0.0</td>
<td>1.52571</td>
<td>0.04623</td>
</tr>
<tr>
<td>140.0</td>
<td>169.71100</td>
<td>-0.22841</td>
<td>0.16667</td>
<td>1.15539</td>
<td>0.0</td>
<td>1.52972</td>
<td>0.05520</td>
</tr>
<tr>
<td>150.0</td>
<td>164.91200</td>
<td>-0.19754</td>
<td>0.16667</td>
<td>1.19219</td>
<td>0.0</td>
<td>1.53381</td>
<td>0.06224</td>
</tr>
<tr>
<td>160.0</td>
<td>160.17100</td>
<td>-0.17125</td>
<td>0.16667</td>
<td>1.23082</td>
<td>0.0</td>
<td>1.53798</td>
<td>0.06773</td>
</tr>
<tr>
<td>170.0</td>
<td>155.49000</td>
<td>-0.14872</td>
<td>0.16667</td>
<td>1.27137</td>
<td>0.0</td>
<td>1.54222</td>
<td>0.07194</td>
</tr>
<tr>
<td>180.0</td>
<td>150.86900</td>
<td>-0.12930</td>
<td>0.16667</td>
<td>1.31398</td>
<td>0.0</td>
<td>1.54655</td>
<td>0.07511</td>
</tr>
</tbody>
</table>

### TABLE 6.5 COMPENSATOR ELEMENTS FOR CONSTANT PRESSURE

<table>
<thead>
<tr>
<th>$E_d$</th>
<th>$P_d$</th>
<th>$D_{12}$</th>
<th>$D_{13}$</th>
<th>$D_{21}$</th>
<th>$D_{23}$</th>
<th>$D_{31}$</th>
<th>$D_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>140.0</td>
<td>-2.27968</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.73343</td>
</tr>
<tr>
<td>20.0</td>
<td>140.0</td>
<td>-1.12972</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.36004</td>
</tr>
<tr>
<td>30.0</td>
<td>140.0</td>
<td>-0.74641</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.16357</td>
</tr>
<tr>
<td>40.0</td>
<td>140.0</td>
<td>-0.55475</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.09234</td>
</tr>
<tr>
<td>50.0</td>
<td>140.0</td>
<td>-0.43975</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.04960</td>
</tr>
<tr>
<td>60.0</td>
<td>140.0</td>
<td>-0.36309</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.02111</td>
</tr>
<tr>
<td>70.0</td>
<td>140.0</td>
<td>-0.30833</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>-0.00076</td>
</tr>
<tr>
<td>80.0</td>
<td>140.0</td>
<td>-0.26726</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.01451</td>
</tr>
<tr>
<td>90.0</td>
<td>140.0</td>
<td>-0.23532</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.02638</td>
</tr>
<tr>
<td>100.0</td>
<td>140.0</td>
<td>-0.20976</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.03588</td>
</tr>
<tr>
<td>110.0</td>
<td>140.0</td>
<td>-0.18886</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.04365</td>
</tr>
<tr>
<td>120.0</td>
<td>140.0</td>
<td>-0.17143</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.05012</td>
</tr>
<tr>
<td>130.0</td>
<td>140.0</td>
<td>-0.15669</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.05560</td>
</tr>
<tr>
<td>140.0</td>
<td>140.0</td>
<td>-0.14405</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.06030</td>
</tr>
<tr>
<td>150.0</td>
<td>140.0</td>
<td>-0.13310</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.06437</td>
</tr>
<tr>
<td>160.0</td>
<td>140.0</td>
<td>-0.12352</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.06793</td>
</tr>
<tr>
<td>170.0</td>
<td>140.0</td>
<td>-0.11506</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.07107</td>
</tr>
<tr>
<td>180.0</td>
<td>140.0</td>
<td>-0.10755</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.07387</td>
</tr>
</tbody>
</table>
### TABLE 6.6 COMPENSATOR ELEMENTS FOR SLIDING PRESSURE

<table>
<thead>
<tr>
<th>$E_d$</th>
<th>$P_d$</th>
<th>$D_{12}$</th>
<th>$D_{13}$</th>
<th>$D_{21}$</th>
<th>$D_{23}$</th>
<th>$D_{31}$</th>
<th>$D_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>65.0</td>
<td>-0.40240</td>
<td>0.16667</td>
<td>3.29411</td>
<td>0.0</td>
<td>1.67042</td>
<td>-0.11107</td>
</tr>
<tr>
<td>20.0</td>
<td>70.0</td>
<td>-0.23299</td>
<td>0.16667</td>
<td>3.03841</td>
<td>0.0</td>
<td>1.65927</td>
<td>-0.03730</td>
</tr>
<tr>
<td>30.0</td>
<td>75.0</td>
<td>-0.17834</td>
<td>0.16667</td>
<td>2.81819</td>
<td>0.0</td>
<td>1.64894</td>
<td>-0.01062</td>
</tr>
<tr>
<td>40.0</td>
<td>80.0</td>
<td>-0.15241</td>
<td>0.16667</td>
<td>2.62662</td>
<td>0.0</td>
<td>1.63931</td>
<td>0.00428</td>
</tr>
<tr>
<td>50.0</td>
<td>85.0</td>
<td>-0.13799</td>
<td>0.16667</td>
<td>2.45852</td>
<td>0.0</td>
<td>1.63030</td>
<td>0.01447</td>
</tr>
<tr>
<td>60.0</td>
<td>90.0</td>
<td>-0.12934</td>
<td>0.16667</td>
<td>2.30988</td>
<td>0.0</td>
<td>1.62183</td>
<td>0.02230</td>
</tr>
<tr>
<td>70.0</td>
<td>95.0</td>
<td>-0.12401</td>
<td>0.16667</td>
<td>2.17753</td>
<td>0.0</td>
<td>1.61384</td>
<td>0.02878</td>
</tr>
<tr>
<td>80.0</td>
<td>100.0</td>
<td>-0.12076</td>
<td>0.16667</td>
<td>2.05897</td>
<td>0.0</td>
<td>1.60629</td>
<td>0.03441</td>
</tr>
<tr>
<td>90.0</td>
<td>105.0</td>
<td>-0.11890</td>
<td>0.16667</td>
<td>1.95218</td>
<td>0.0</td>
<td>1.59913</td>
<td>0.03948</td>
</tr>
<tr>
<td>100.0</td>
<td>110.0</td>
<td>-0.11804</td>
<td>0.16667</td>
<td>1.85551</td>
<td>0.0</td>
<td>1.59231</td>
<td>0.04415</td>
</tr>
<tr>
<td>110.0</td>
<td>115.0</td>
<td>-0.11790</td>
<td>0.16667</td>
<td>1.76760</td>
<td>0.0</td>
<td>1.58582</td>
<td>0.04853</td>
</tr>
<tr>
<td>120.0</td>
<td>120.0</td>
<td>-0.11831</td>
<td>0.16667</td>
<td>1.68733</td>
<td>0.0</td>
<td>1.57962</td>
<td>0.05268</td>
</tr>
<tr>
<td>130.0</td>
<td>125.0</td>
<td>-0.11916</td>
<td>0.16667</td>
<td>1.61375</td>
<td>0.0</td>
<td>1.57369</td>
<td>0.05667</td>
</tr>
<tr>
<td>140.0</td>
<td>130.0</td>
<td>-0.12035</td>
<td>0.16667</td>
<td>1.54608</td>
<td>0.0</td>
<td>1.56801</td>
<td>0.06053</td>
</tr>
<tr>
<td>150.0</td>
<td>135.0</td>
<td>-0.12182</td>
<td>0.16667</td>
<td>1.48363</td>
<td>0.0</td>
<td>1.56255</td>
<td>0.06427</td>
</tr>
<tr>
<td>160.0</td>
<td>140.0</td>
<td>-0.12352</td>
<td>0.16667</td>
<td>1.42584</td>
<td>0.0</td>
<td>1.55730</td>
<td>0.06793</td>
</tr>
<tr>
<td>170.0</td>
<td>145.0</td>
<td>-0.12542</td>
<td>0.16667</td>
<td>1.37221</td>
<td>0.0</td>
<td>1.55225</td>
<td>0.07151</td>
</tr>
<tr>
<td>180.0</td>
<td>150.0</td>
<td>-0.12748</td>
<td>0.16667</td>
<td>1.32231</td>
<td>0.0</td>
<td>1.54738</td>
<td>0.07504</td>
</tr>
</tbody>
</table>

### TABLE 6.7 COMPENSATOR ELEMENTS FOR VARIABLE PRESSURE

<table>
<thead>
<tr>
<th>$E_d$</th>
<th>$P_d$</th>
<th>$D_{12}$</th>
<th>$D_{13}$</th>
<th>$D_{21}$</th>
<th>$D_{23}$</th>
<th>$D_{31}$</th>
<th>$D_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>32.00100</td>
<td>-0.07951</td>
<td>0.16667</td>
<td>7.12688</td>
<td>0.0</td>
<td>1.77925</td>
<td>-0.00940</td>
</tr>
<tr>
<td>20.0</td>
<td>38.25950</td>
<td>-0.05724</td>
<td>0.16667</td>
<td>5.86786</td>
<td>0.0</td>
<td>1.75143</td>
<td>0.00602</td>
</tr>
<tr>
<td>30.0</td>
<td>44.51910</td>
<td>-0.05234</td>
<td>0.16667</td>
<td>4.97545</td>
<td>0.0</td>
<td>1.72803</td>
<td>0.01362</td>
</tr>
<tr>
<td>40.0</td>
<td>50.77860</td>
<td>-0.05186</td>
<td>0.16667</td>
<td>4.31123</td>
<td>0.0</td>
<td>1.70787</td>
<td>0.01925</td>
</tr>
<tr>
<td>50.0</td>
<td>57.03810</td>
<td>-0.05321</td>
<td>0.16667</td>
<td>3.79833</td>
<td>0.0</td>
<td>1.69017</td>
<td>0.02408</td>
</tr>
<tr>
<td>60.0</td>
<td>63.29760</td>
<td>-0.05552</td>
<td>0.16667</td>
<td>3.39080</td>
<td>0.0</td>
<td>1.67442</td>
<td>0.02850</td>
</tr>
<tr>
<td>70.0</td>
<td>69.55720</td>
<td>-0.05843</td>
<td>0.16667</td>
<td>3.05950</td>
<td>0.0</td>
<td>1.66022</td>
<td>0.03267</td>
</tr>
<tr>
<td>80.0</td>
<td>75.81670</td>
<td>-0.06173</td>
<td>0.16667</td>
<td>2.78509</td>
<td>0.0</td>
<td>1.64732</td>
<td>0.03670</td>
</tr>
<tr>
<td>90.0</td>
<td>82.07620</td>
<td>-0.06533</td>
<td>0.16667</td>
<td>2.55422</td>
<td>0.0</td>
<td>1.63550</td>
<td>0.04061</td>
</tr>
<tr>
<td>100.0</td>
<td>88.33580</td>
<td>-0.06914</td>
<td>0.16667</td>
<td>2.35740</td>
<td>0.0</td>
<td>1.62459</td>
<td>0.04444</td>
</tr>
<tr>
<td>110.0</td>
<td>94.59530</td>
<td>-0.07315</td>
<td>0.16667</td>
<td>2.18770</td>
<td>0.0</td>
<td>1.61447</td>
<td>0.04822</td>
</tr>
<tr>
<td>120.0</td>
<td>100.85480</td>
<td>-0.07720</td>
<td>0.16667</td>
<td>2.03993</td>
<td>0.0</td>
<td>1.60504</td>
<td>0.05194</td>
</tr>
<tr>
<td>130.0</td>
<td>107.11440</td>
<td>-0.08159</td>
<td>0.16667</td>
<td>1.91015</td>
<td>0.0</td>
<td>1.59620</td>
<td>0.05563</td>
</tr>
<tr>
<td>140.0</td>
<td>113.37390</td>
<td>-0.08599</td>
<td>0.16667</td>
<td>1.79530</td>
<td>0.0</td>
<td>1.58790</td>
<td>0.05928</td>
</tr>
<tr>
<td>150.0</td>
<td>119.63340</td>
<td>-0.09049</td>
<td>0.16667</td>
<td>1.69297</td>
<td>0.0</td>
<td>1.58007</td>
<td>0.06290</td>
</tr>
<tr>
<td>160.0</td>
<td>125.89290</td>
<td>-0.09509</td>
<td>0.16667</td>
<td>1.60125</td>
<td>0.0</td>
<td>1.57266</td>
<td>0.06650</td>
</tr>
<tr>
<td>170.0</td>
<td>132.15250</td>
<td>-0.09978</td>
<td>0.16667</td>
<td>1.51859</td>
<td>0.0</td>
<td>1.56563</td>
<td>0.07007</td>
</tr>
<tr>
<td>180.0</td>
<td>138.41200</td>
<td>-0.10455</td>
<td>0.16667</td>
<td>1.44372</td>
<td>0.0</td>
<td>1.55895</td>
<td>0.07363</td>
</tr>
</tbody>
</table>
### TABLE 6.8 COMPENSATOR ELEMENTS FOR LOW-PRESSURE LIMIT

<table>
<thead>
<tr>
<th>$E_d$</th>
<th>$P_d$</th>
<th>$D_{12}$</th>
<th>$D_{13}$</th>
<th>$D_{21}$</th>
<th>$D_{23}$</th>
<th>$D_{31}$</th>
<th>$D_{32}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0</td>
<td>32.00100</td>
<td>-0.07951</td>
<td>0.16667</td>
<td>7.12688</td>
<td>0.0</td>
<td>1.77925</td>
<td>-0.00940</td>
</tr>
<tr>
<td>20.0</td>
<td>32.00100</td>
<td>-0.03744</td>
<td>0.16667</td>
<td>7.12688</td>
<td>0.0</td>
<td>1.77925</td>
<td>0.00917</td>
</tr>
<tr>
<td>30.0</td>
<td>32.00100</td>
<td>-0.02342</td>
<td>0.16667</td>
<td>7.12688</td>
<td>0.0</td>
<td>1.77925</td>
<td>0.01536</td>
</tr>
<tr>
<td>40.0</td>
<td>36.34100</td>
<td>-0.02272</td>
<td>0.16667</td>
<td>6.20580</td>
<td>0.0</td>
<td>1.75942</td>
<td>0.01934</td>
</tr>
<tr>
<td>50.0</td>
<td>44.31400</td>
<td>-0.02850</td>
<td>0.16667</td>
<td>5.00053</td>
<td>0.0</td>
<td>1.72874</td>
<td>0.02358</td>
</tr>
<tr>
<td>60.0</td>
<td>52.11000</td>
<td>-0.03429</td>
<td>0.16667</td>
<td>4.19136</td>
<td>0.0</td>
<td>1.70392</td>
<td>0.02773</td>
</tr>
<tr>
<td>70.0</td>
<td>59.76300</td>
<td>-0.04010</td>
<td>0.16667</td>
<td>3.60998</td>
<td>0.0</td>
<td>1.68310</td>
<td>0.03180</td>
</tr>
<tr>
<td>80.0</td>
<td>67.29500</td>
<td>-0.04591</td>
<td>0.16667</td>
<td>3.17182</td>
<td>0.0</td>
<td>1.66519</td>
<td>0.03581</td>
</tr>
<tr>
<td>90.0</td>
<td>74.72200</td>
<td>-0.05174</td>
<td>0.16667</td>
<td>2.82963</td>
<td>0.0</td>
<td>1.64949</td>
<td>0.03976</td>
</tr>
<tr>
<td>100.0</td>
<td>82.05800</td>
<td>-0.05758</td>
<td>0.16667</td>
<td>2.55484</td>
<td>0.0</td>
<td>1.63553</td>
<td>0.04366</td>
</tr>
<tr>
<td>110.0</td>
<td>89.31300</td>
<td>-0.06342</td>
<td>0.16667</td>
<td>2.32927</td>
<td>0.0</td>
<td>1.62296</td>
<td>0.04752</td>
</tr>
<tr>
<td>120.0</td>
<td>96.49500</td>
<td>-0.06927</td>
<td>0.16667</td>
<td>2.14074</td>
<td>0.0</td>
<td>1.61154</td>
<td>0.05134</td>
</tr>
<tr>
<td>130.0</td>
<td>103.61100</td>
<td>-0.07512</td>
<td>0.16667</td>
<td>1.98077</td>
<td>0.0</td>
<td>1.60108</td>
<td>0.05513</td>
</tr>
<tr>
<td>140.0</td>
<td>110.66600</td>
<td>-0.08098</td>
<td>0.16667</td>
<td>1.84332</td>
<td>0.0</td>
<td>1.59143</td>
<td>0.05888</td>
</tr>
<tr>
<td>150.0</td>
<td>117.66500</td>
<td>-0.08684</td>
<td>0.16667</td>
<td>1.72393</td>
<td>0.0</td>
<td>1.58248</td>
<td>0.06261</td>
</tr>
<tr>
<td>160.0</td>
<td>124.61200</td>
<td>-0.09271</td>
<td>0.16667</td>
<td>1.61924</td>
<td>0.0</td>
<td>1.57414</td>
<td>0.06630</td>
</tr>
<tr>
<td>170.0</td>
<td>131.51200</td>
<td>-0.09859</td>
<td>0.16667</td>
<td>1.52667</td>
<td>0.0</td>
<td>1.56633</td>
<td>0.06997</td>
</tr>
<tr>
<td>180.0</td>
<td>138.36600</td>
<td>-0.10446</td>
<td>0.16667</td>
<td>1.44425</td>
<td>0.0</td>
<td>1.55900</td>
<td>0.07362</td>
</tr>
</tbody>
</table>
6.5 Summary

The feedback control processor module of the ICCS-MP consist of a multiloop scheme consisting of fuzzy-PID controllers and a multivariable interaction compensator, whose design and adjustment is carried out at the supervisory level. This chapter presented the design of the controllers and the compensator, as well as their scheduling strategy. Because of its nature, the description of the controller tuning system at the supervisory level is postponed to the next chapter, after all the components of the ICCS-MP are integrated. The design of the feedback control processor module allows the whole control system structure to be easily expanded or contracted as required by the scope of coordination of the ICCS. The 2D scheduling approach allows the feedback module to be automatically customized to any change in the power-pressure operating policy, thus providing the whole control system with adaptability characteristics, some of them active during on-line operation to provide wide-range operation along the whole power unit operating range, and others are carried out off-line across the pressure operating range to provide for process optimization.
CHAPTER 7

INTELLIGENT COORDINATED CONTROL

In this chapter, the ICCS-MP gets fully integrated from the components described in the previous chapters, and its performance is showed based on simulation experiments. In Section 7.1 the whole integration of the ICCS-MP is described. In Section 7.2 the remaining component applications, not described in previous chapters, are presented. These include the Genetic Algorithm-based fuzzy-PID tuner, the performance monitor, and the operating state monitor. As motivation and as a reference, Section 7.3 shows the most relevant control related problems faced by a conventional coordinated control scheme, which are to be solved by the ICCS-MP. In Section 7.4, system partial tests aimed to expose the role and contribution of each component of the ICCS-MP to the system total response are presented. In Section 7.5, integral system test are carried out to demonstrate the characteristics of the whole system. Finally, in Section 7.6 conclusions on the test results are drawn.

7.1 Introduction

In Chapter 3, the ICCS-MP was proposed as a prototype realization to demonstrate the feasibility of the ICCS paradigm to provide an answer to the current power generation context faced by fossil-fuel power units. In this regard, the ICCS-MP was designed to capture the major characteristics of the ICCS. Basically, the ICCS-MP provides the means to achieve optimal wide-range cyclic operation of a power unit within the context of a totally automated power system. That is, the ICCS-MP allows a plant to follow any given unit load demand profile, issued by economic dispatch and unit commitment applications, with optimal accommodation of any operation scenario
characterized by an arbitrary number of, generally conflicting, operating objectives. From a wider perspective, the ICCS-MP constitutes a minimal realization of a generalized overall unit control system for FFPUs that satisfies three main design requirements: related to current control practice, open control structure, and structured system organization. First, the ICCS-MP constitutes an extension to the current concept of coordinated control in power units. In fact, the control structure of the ICCS-MP has been described as an upgrade of a conventional coordinated control strategy in several steps. Second, the proposed open control structure allows contraction or expansion to release or to include control loops into the base control coordination strategy. This is exemplified in the ICCS-MP by including the drum water level control loop into the basic coordination strategy. Third, beyond the openness of the control strategy, openness of the whole control system is further supported by an agent-based design that will provide even more versatility to the ICCS-MP. Nevertheless, the main concern is to deal systematically and since the beginning with the complexity of the future software implementation required to achieve such versatility. As described, both wide and narrow points of view of the ICCS-MP are necessary and complementary. The wide sense viewpoint corresponds to the conceptual model and the narrow sense viewpoint corresponds to the working device that proves the correctness of the concept.

The ICCS-MP implements an open two-level hierarchical intelligent hybrid multi-agent coordinated control system as defined in Section 3.4. The functional block diagram of the ICCS-MP shows the component agents (Figure 3.8). The physical scope diagram of the ICCS-MP (Figure 3.9) shows the context of the control strategy as applied to a FFPU and provides the main input and output signals of the system. In a first level of detail, the components of the ICCS-MP were grouped in three main modules: reference governor, feedforward control processor, and feedback control processor (Figure 3.10). At this level, a 2 degrees-of-freedom control structure is implemented. A second level of detail shows the location of the component agents as related to the control structure (Figure 3.11). This view also summarizes the techniques involved in the development of the applications. Finally, the details of the control strategy as an extended coordinated
control scheme are shown in Figure 3.13, which clearly shows the components associated with the optimal set-point scheduling, reference-feedforward compensation, control loop interaction (disturbance-feedforward) compensation, and closed-loop feedback control.

From a methodological point of view, the development of the ICCS-MP as provided in this thesis constitutes itself an application of the proposed methodology to design and develop a generalized overall unit control system for a FFPU. This methodology follows a software process life-cycle of the rapid-prototyping type that is directly reflected in the organization of this dissertation. Chapters 1 and 2 constitute the specification of requirements; mainly from the areas of control engineering and power plant process engineering. Chapter 3 constitutes the design stage. It describes the conceptual model of the ICCS as a multiagent system, and specifies a minimum prototype (ICCS-MP) for realization. Chapters 4, 5, and 6 undertake the implementation and validation (module testing) of the ICCS-MP main components: reference governor, feedforward control processor, and feedback control processor. Finally, Chapter 7 describes the integration and validation (system testing) of the ICCS-MP.

Of all required applications to integrate the ICCS-MP it remains to introduce the genetic algorithm-based tuner, the power unit performance monitor, and the operating state monitor, which are described in Section 7.2. Then, the integration and validation of the ICCS-MP takes place in two parts. First, starting from a conventional coordinated control system as currently available at FFPU, all components are successively incorporated and validated in a set of system partial tests that includes the set-point scheduler, feedforward control, fuzzy-PID scheduling, and interaction compensator scheduling. Second, once all components are in place a full set of system integral test is carried out. This set includes tests to show the behavior of the ICCS-MP during unit load demand steps, load ramps, cyclic operation, and measurement noise.

The control system is embedded in a simulation shell for the analysis of overall unit control. The shell was developed in a personal computer platform using the Matlab/Simulink programming environment. A set of 50+ tests are available to fully characterize the process and to tune and analyze the control system. Tests can be
executed at various levels. At the unit model level test can be carried out to verify the open-loop process model responses. At the direct level the models of the control valve actuators are included to impose magnitude and rate of change limitations on the control signals to the process model. At the master level the control algorithms are included. At the supervisory level the set-point scheduler is included. Finally, at the system level any unit load demand can be specified to carry out cyclic operation tests comprehending the whole control system and all the process models.

### 7.2 Ancillary Applications

Of all the components comprehended by the ICCS-MP that were specified in Section 3.4, it remains to present the feedback control tuning and process monitoring applications. While the former was delayed until this chapter because of its nature requiring all the control related components already incorporated, the later were kept as very simple applications regarding on-line monitoring of the controlled FFPU performance and its operating state. The monitoring applications may certainly become very important and sophisticated applications when functions such as fault diagnosis, test assistance, model building, protections, etc., are also considered in the control system, but for now they are out of the scope of the ICCS-MP.

#### 7.2.1 Tuning of PID based multiloop control systems

In contrast to the SISO case, the interactions among the control loops of a multiloop control system pose the highest difficulty to proper controller tuning. The obvious approach to tune the controllers individually and then expect that the system performance will be adequate when all the loops are eventually closed is not realistic, particularly when strong interactions are present as is the case of the FFPU. Despite the
abundance of PID-based decentralized control systems, the availability of multivariable tuning methods is still scarce. Among the most well-known methods is an extension to the multivariable case of the Ziegler-Nichols method proposed by [Niederlinski 1971]. Partial extensions of the single loop relay auto-tuner to the MIMO case have been proposed by [Zgorzelki, et al. 1990, Loh, et al. 1990]. A full extension of the auto-tuner to the MIMO case was proposed by [Halevi, et al. 1997]. In all cases, it is not clear and it is not specified in what sense the parameters provided by these methods are good. In addition, the procedures are rather involved and as will be shortly shown provide values valid for a small neighborhood about the tuning operating point, which is adequate for regulation purposes but not for wide-range operation as required. Because of these reasons, the procedure followed to tune the controllers in this work is similar to that typically followed in practice:

- Tune each control loop independently until satisfactory closed-loop step response performance is obtained, with the other loops on manual control mode or in a rather loose proportional control.
- Set all control-loops to automatic control mode (closed-loop) and retune the control parameters until the over all closed-loop performance is satisfactory in all the loops.

It should be noted that for systems with significant control loop interaction, as is the case for the FFPU, the second step can be very difficult and tedious. A rule of thumb that can reduce the retuning time is that the controllers will need to be made more conservative when all the loops are closed, that is, in most cases the proportional gains should be reduced and the integral gains decreased. In particular, the following procedure was used to tune the controller parameters (proportional gain, $K_p$, integral gain, $K_i$, and derivative gain, $K_d$) at a given operating point through step responses of 1 MW, 1 kg/cm$^2$, and 5 mm in the power, pressure, and level set-points loops, respectively:

- Around a known equilibrium point (e.g., $E=80$ MW, $P=100$ kg/cm$^2$, and $L=0$ mm), obtain a stable response with small proportional gains only in the three control loops.
- Increase $K_p$ in the power loop as much as an stable response is obtained, then do the same with $K_p$ in the pressure loop, and finally with $K_p$ in the level loop.
• Review the $K_p$ in the three control loops together to obtain a stable response with acceptable response speed.

• Increase $K_i$ in the power loop to eliminate the steady-state error in a reasonable amount of time without too much oscillation, then do the same with $K_i$ in the pressure loop, and finally with $K_i$ in the level loop.

• Review the $K_i$ in the three control loops together to obtain an acceptable settlement time and oscillation.

• Increase $K_d$ in the power loop to attenuate the oscillation without degrading too much the speed of response, then do the same with $K_d$ in the pressure loop.

• Increase $K_i$ in the level loop to correct the speed of the level response affected by the attenuation in the power and pressure responses in the previous step. This may again increase the oscillation in the power and pressure responses.

• Review $K_d$ in the pressure control loop to decrease any oscillation in pressure and power.

After this process has been completed, it could still be necessary to fine tune the proportional gains in all the controllers again. As a word of caution, and as will be shortly shown, it is not recommended to try to achieve critically damped step responses in the three control loops because this degrades the wide-range load-tracking ramp response of the FFPU; instead a certain degree of slackness (underdamped response) is required.

As can be expected from the lengthy procedure above, tuning the control system is a very tedious and time consuming process, which simply becomes overwhelming as the number of operating points at which tuning is required increases, as is the case for the scheduling strategy in this research. To assist this task, parameter tuning is carried out via simulation using an optimization procedure based on the genetic algorithm (GA) paradigm [Holland 1992]. The implementation of the GA-based tuning application in the ICCS-MP allows optimization of one or more parameters, as required, and it is used off-line to speed up the parameter tuning by applying it sequentially through the steps of the previously described tuning procedure.
7.2.2 Genetic algorithm-based tuning

Genetic algorithms (GAs) are global search methods that emulate the process of biological evolution [Man, et al. 1997]. A GA operates on a population of initial solution estimates (individuals). At each stage (generation) of the genetic cycle a new set of approximate solutions is generated by means of genetic operations (reproduction, mutation, and recombination). Some of these intermediate solutions are selected, according to their level of fitness in the problem domain, to continue evolving (survival of the fittest). This repeated evolution process leads to a set of solutions that are better suited to their environment than the solution from where they were created.

Table 7.1 outlines the pseudocode of a simple genetic algorithm [Goldberg 1989]. In preparation for optimization by GA, individuals are represented as encoded strings of characters (chromosomes) in such a way that the values of the chromosomes (genotypes) are uniquely mapped onto the decision variables (phenotypes). Then, the GA basically consist of two phases: initialization and evolutionary cycle.

<table>
<thead>
<tr>
<th>TABLE 7.1 BASIC GENETIC ALGORITHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure GA {</td>
</tr>
<tr>
<td>t=0;</td>
</tr>
<tr>
<td>initialize P(t);</td>
</tr>
<tr>
<td>evaluate P(t);</td>
</tr>
<tr>
<td>do {</td>
</tr>
<tr>
<td>t=t+1;</td>
</tr>
<tr>
<td>select P(t) from P(t-1);</td>
</tr>
<tr>
<td>recombine P(t);</td>
</tr>
<tr>
<td>mutate P(t);</td>
</tr>
<tr>
<td>evaluate P(t);</td>
</tr>
<tr>
<td>} while(termination condition not satisfied)</td>
</tr>
<tr>
<td>} }</td>
</tr>
</tbody>
</table>
In the initialization phase the population $P(t)$ is initialized by randomly sampling the chromosome space. Then, the initial population is evaluated by converting the chromosomes to their phenotypic values and evaluating an appropriate objective function. The objective function characterizes the performance of a given individual in the optimization problem domain, that is, defines the relative fitness of an individual in the evolution process.

Once the initialization phase of the GA has been accomplished, the cyclic evolutionary phase takes place. Each iteration of the cycle consists of four steps. The selection step is first. Highly fit individuals, relative to the whole population, have a high probability of being selected for reproduction, while less fit individuals have a lower probability of being selected. In the second and third step the selected individuals are modified through the application of genetic operators to produce the next generation. Genetic operators manipulate the characters (genes) of the chromosomes directly, under the assumption that certain individual’s gene codes, on average, produce fitter individuals. Genetic operators may be divided into two main categories: recombination (or crossover) and mutation. During recombination, at the second step, genetic information is exchanged between pairs, or larger groups, of individuals in order to create new chromosomes. Recombination is not necessarily applied to all groups of individuals selected for reproduction, allowing some chromosomes to survive into successive generations unaltered. Mutation, executed in the third step, alters the genetic representations of some individuals according to a given probabilistic rule. Generally, mutation is applied with a much lower probability than recombination and is considered as a background operator that ensures that the probability of searching any particular subspace of the search space is never zero. In the fourth step, after recombination and mutation, the individual strings are decoded and evaluated by the objective function, thus ending one iteration of the evolutionary cycle.

The evolutionary process continues through subsequent iterations. At each iteration the average performance of individuals in a population is expected to increase as fit individuals are preserved and reproduced with one another and less fit individuals die
out. The GA is terminated when a given criterion (number of generations, mean deviation in the average performance of the population, some point of the search space is encountered, etc.) is satisfied.

Application of the GA to the optimization of the PID controller parameters was done by converting the $K_p$, $K_i$, and $K_d$ parameters to unsigned binary code with a length of 12 bits using the relationship:

$$b = \frac{2^{12} - 1}{V_{up} - V_{lo}} (V - V_{lo})$$  \hspace{1cm} (7.1)

where $b$ is the rounded binary value of the decimal $V$ (phenotype) in the range $[V_{lo}, V_{up}]$. Depending on the step of the tuning process, up to three parameters are concatenated to form a $3 \times 12 = 36$ bit binary string (chromosome) such as:

$$S: \begin{array}{c|c|c|c}
K_p & K_i & K_d \\
100101001101 & 01110100001111 & 110010100100 \\
\end{array}$$  \hspace{1cm} (7.2)

The fitness of each possible solution $S$ is obtained by evaluating a fitness function, which can be defined in terms of a performance index. In this case the fitness function is a simple function of the form:

$$f(J) = \frac{1}{1 + J}$$  \hspace{1cm} (7.3)

where $J$ is any adequate performance index of the system response, for instance the IAE index of the error response, $e(t)$, of the controlled system output to a set-point step, from the initial time $t_i$ to the final time $t_f$:

$$J = \int_{t_i}^{t_f} |e(t)| dt$$  \hspace{1cm} (7.4)

With these definitions, the implementation of the GA proceeds as follows:

1) Produce an initial population of randomly generated sets of parameters $(K_p, K_i, K_d)$.
2) Simulate the step response of the closed-loop system and evaluate the fitness function of each set of parameters.
3) Select by the roulette wheel method the sets of parameters to be reproduced.
4) Recombine by pairs the selected sets of parameters with a randomly generated point of crossover.

5) Mutate the selected sets of parameters according to the probability of mutation.

6) Repeat steps 2) through 5) iteratively until the fittest set of parameters provides and acceptable closed loop response.

Note that steps 1) and 2) are carried out with the decimal representation of the controller parameters (phenotype), while steps 3), 4) and 5) are carried out with the binary representation in a binary string (chromosome). Also, the genetic operations mentioned in steps 3), 4) and 5) correspond to those of the Simple Genetic Algorithm as proposed in [Goldberg 1989]. The results of the application of the GA-based tuning procedure are presented in some of the sections that follow, as required.

7.2.3 Performance monitor

The performance monitor is a simple application that evaluates different performance indices and objective functions during the system operation (simulated). The indices and objective functions include:

\[
\text{IAE} = \int_{t_i}^{t_f} |e(t)| dt \tag{7.5a}
\]

\[
\text{ITAE} = \int_{t_i}^{t_f} t |e(t)| dt \tag{7.5b}
\]

\[
\text{ISE} = \int_{t_i}^{t_f} e^2(t) dt \tag{7.5c}
\]

\[
\text{ITSE} = \int_{t_i}^{t_f} te^2(t) dt \tag{7.5d}
\]

\[
J_1 = \int_{t_i}^{t_f} |E_d - E| dt \tag{7.5e}
\]

\[
J_2 = \int_{t_i}^{t_f} u_i(t) dt \tag{7.5f}
\]
\[ J_3 = -\int_{t_i}^{t_f} u_2(t) \, dt \]  
\[ J_4 = -\int_{t_i}^{t_f} u_3(t) \, dt \]

where \( J_1, J_2, J_3 \) and \( J_4 \) are the accumulated values of the objective functions defined in Section 4.5.1 to design the multiobjective optimal set-point mappings for process optimization.

### 7.2.4 Operating state monitor

The operating state monitor is another simple application that continually checks the operating state of the FFPU during operation (simulated). For the case of this research work, it monitors the deviation between the commanded set-point trajectories and the actual outputs to issue a permissive flag to indicate operation in the normal state when these errors are small. For the variables being considered in this dissertation a ±5% variation about the commanded value is considered adequate. The relevance of this information is that it validates the conditions to satisfy the small deviation requirement to preserve the characteristics of a gain scheduling control system between the selected scheduling points.
7.3 Conventional Coordinated Control

As mentioned earlier, the control scheme of the ICCS-MP may be regarded as an upgrade of a conventional coordinated control scheme. So, in this section a closer look is taken into the performance of a typical CC scheme. Results obtained in this section motivate and will be used as a reference for comparison of the ICCS-MP performance.

At fossil fuel power units, the CC scheme constitutes the uppermost layer of the control system. The CC is responsible for driving the boiler-turbine-generator set as a single entity. Also, the CC constitutes the primary means to achieve process optimization through control. The dominant behavior of the unit is governed through the power and steam pressure control loops. Given a unit load demand, the CC provides control signals to the boiler and to the steam turbine to match the responses of the boiler and the turbine-generator during load changes and load disturbances. Ordinarily, the set-point for the pressure control loop is obtained from the unit load demand through a power-pressure non-linear mapping along the whole power operating-range of the unit. The configuration of the typical CC scheme under consideration was shown in Figure 1.2, as corresponds to a coordinated turbine-follower mode [Landis and Wulfsohn 1988]. The power controller generates commands for the fuel/air valve positions, $u_1$, from the measured generated power, $E$, and the power demand, $E_{d}$, which is equal to the unit load demand, $E_{uld}$. The pressure controller drives the throttle valve calculating the position demand, $u_2$, from the measured steam pressure, $P$, and the pressure set-point, $P_{d}$, which is obtained from the unit load demand, $E_{uld}$, through a non-linear power-pressure mapping.

As for most control systems in the process industries, the CC scheme in a FFPU consists of a decentralized multiloop configuration of single-input-single-output feedback control loops evaluating conventional PI or PID algorithms. Despite its simple structure, decentralized PID control has a long record of satisfactory performance; its effectiveness to regulate the process under random load disturbances around a fixed operating point is daily proven all over the world. The main reason for this is the relatively simple structure of the control system, which is easy to understand and to implement, and its reliability in
case of actuator or sensor failure, that could make it relatively easy to manually stabilize a system when only one loop is directly affected. In addition, the number of tuning parameters is considerably small, and increases only linearly with the dimension of the process (i.e., 3n tuning parameters for a control system with n control loops).

Normally, the controller parameters are tuned at some predefined operating point (i.e., base load) assuming nearly constant load conditions, and are left fixed thereafter. This approach works well for process regulation about the operating point used for tuning. However, current requirements demanding wide-range operation of FFPUs challenge this approach. The performance of the power unit may decrease due to the nonlinear and interactive dynamics of the process that change with the current operating point. As a consequence, strong physical demands that are detrimental of the unit duty life may be imposed on the plant equipment. In the subsections that follow the drawbacks of a the typical CC, considered as a PID-based multiloop control system, are shown through simulation experiments. The results obtained are later used as a reference for comparison and to evaluate the performance of the ICCS-MP, which thus provides a feasible solution to the optimal wide-range operation requirements of FFPUs.

### 7.3.1 Control loop interaction and tuning

The main difficulty for decentralized control of multivariable processes is that of tuning the controllers because of control loop interaction due to the coupling dynamics among the process inputs and outputs. Interaction among the main inputs and outputs of a FFPU was shown in Chapter 2, through the open-loop response to steps in the control valve demands. Now, the effects of control loop interaction for the same FFPU are shown through the closed-loop response to steps in the set-points. Figure 7.1 shows the FFPU response to a unit step in the power set-point. Figure 7.2 shows the FFPU response to a unit step in the steam pressure set-point. Finally, Figure 7.3 exhibits the response to a 5 unit step response in the level set-point.
Figure 7.1 Response to power set-point step.
Figure 7.2 Response to pressure set-point step.
Figure 7.3  Response to level set-point step.
In each figure, the graphs in the left column present the FFPU response to the steps, and the graphs in the right column show the behavior of the associated control signals. Each row corresponds to the variables paired through the control loop. All tests were carried out starting from an operating point at half load, defined by $E=80$ MW, $P=100$ kg/cm$^2$, and $L=0$ mm. The controllers were tuned to achieve an almost critically damped response in all loops. The controller parameters are summarized in Table 7.2.

**TABLE 7.2  PARAMETERS OF PID CONTROLLERS**

<table>
<thead>
<tr>
<th>Controller</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>0.090</td>
<td>9.0x10$^{-4}$</td>
<td>0.27</td>
</tr>
<tr>
<td>Pressure</td>
<td>0.025</td>
<td>2.5x10$^{-4}$</td>
<td>0.50</td>
</tr>
<tr>
<td>Level</td>
<td>0.008</td>
<td>8.0x10$^{-6}$</td>
<td>-</td>
</tr>
</tbody>
</table>

Results show that the strongest and most disturbing interaction is from the pressure control loop to the power output, which normally is required the tightest control for either tracking or regulation. Second in importance is the interaction from the power loop to the pressure output, which may adversely affect the physical condition of the plant equipment. In third place are the interactions from the power and pressure control loops to the level output, for which a not so tight regulation is usually required provided that the magnitude of the oscillations about the zero level is kept within safe limits. Finally, the interactions from the level control loop to the power and pressure outputs are both relatively small and are usually a minor concern.

### 7.3.2 Wide-range load-tracking

Certainly, nice step responses are an indicator of good control performance. Most control systems of all kinds are usually demonstrated using this approach. Unfortunately, for the case of nonlinear MIMO systems subject to wide-range reference-tracking
operation requirements it is not sufficient to guarantee good performance. In the case of FFPU's there are several practical considerations that prevent the utilization of step responses; ramp responses are preferred. Strictly speaking, ramp responses provide the same amount of information over the system as step responses do. Steps responses will be used in the remaining of this work in to exhibit the behavior of the system solely in simulation experiments, under the understanding that this tests are not usually recommended to be carried out in practice. Now, the ramp response of the FFPU is investigated for a not very demanding ramp-up loading maneuver using the same controller parameters as in Section 7.3.1, which provided excellent step responses. Power is required to increase from 80 MW (half load) to 90 MW in 150 seconds, that is a 6.25% power set-point change with a rate of 2.5 %/min, which would be normally considered an easy test. Accordingly, the pressure set-point is obtained from the unit load demand through the mapping:

\[
P_d = \frac{150 - 65}{180 - 10} F_{u_d} + 65
\]  

(7.6)

that implements a typical sliding-pressure operating policy, which is a fairly common practice in CC schemes. As can be seen from the graphs in Figure 7.4, the power set-point tracking is just good, with an excellent low control activity. Pressure set-point tracking is poor, specially at the end of the ramp with a large overshoot and large settling time, but its control activity is excellent. The oscillations in the water level are fine and the control activity is also excellent. These results demonstrate that controller tuning for excellent step response does not implies good load-tracking performance, which is a major concern for wide-range operation.

The inverse situation is also interesting. Controllers tuned to achieve excellent ramp response do not necessarily provide good step response, or even stable response. This fact is shown by Figures 7.5 through 7.8 for the same load ramp maneuver as before, but with the controller parameters retuned (Table 7.3) to improve the ramp response.
Figure 7.4 Ramp load tracking with step-tuned controller parameters.
Figure 7.5  Ramp load-tracking with ramp-tuned controller parameters.
Figure 7.6 Response to step in power set-point with ramp-tuned parameters.
Figure 7.7  Response to step in pressure set-point with ramp-tuned parameters.
Figure 7.8  Response to step in level set-point with ramp-tuned parameters.
### TABLE 7.3 PARAMETERS OF PID CONTROLLERS FOR RAMP RESPONSE

<table>
<thead>
<tr>
<th>Controller</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>0.090</td>
<td>9.0x10^{-4}</td>
<td>0.27</td>
</tr>
<tr>
<td>Pressure</td>
<td>0.025</td>
<td>2.5x10^{-4}</td>
<td>0.50</td>
</tr>
<tr>
<td>Level</td>
<td>0.008</td>
<td>8.0x10^{-6}</td>
<td>-</td>
</tr>
</tbody>
</table>

7.3.3 Changing operating point

Process optimization under changing operating scenarios will certainly demand operation at different operating points. This situation leads to another drawback of the fixed parameter control configuration. The unit performance will not be the same at different operating points; not even a variation proportional to the change in the operating conditions holds true. Figure 7.9 shows the step responses at the operating point defined by $E=20$ MW, $P=35$ kg/cm$^2$, and $L=0$ mm, using the same controller parameters as in Section 7.3.1. Clearly, process optimization along and across the operating space of the FFPU requires the implementation of a mechanism to cope with this changes.

7.3.4 Process nonlinearity

As expected for a nonlinear system, the effects of process nonlinearity also make the multiloop PID control to miss the mark. Systems settings are usually valid for a relatively small neighborhood around the tuning point. This is exemplified in Figure 7.10 that shows the response to larger steps under the same tuning and initial conditions of Section 7.3.1. Results clearly show that the excellent response in Section 7.3.1 is not preserved due to the process nonlinearity.
Figure 7.9 Responses to set-point steps at different operating point.
Figure 7.10  Response to large set-point steps.
7.4 System Partial Tests

The set of tests in this section demonstrates the integration of the ICCS-MP and show the effect of its components to overcome the drawbacks of the typical CC based on conventional PID controllers and the sliding pressure operating policy. Comparison is made against the results obtained in Section 7.3 using the typical CC.

7.4.1 Feedforward control

An almost immediate way of improving the performance of the conventional CC is to incorporate the fuzzy feedforward control path described in Chapter 5 to the already existing feedback control. This action creates a multivariable two-degrees-of-freedom controller. The feedforward control is responsible to provide the main contribution to the control signals to drive the FFPU along wide-range operation patterns enforcing the pre-specified power-pressure operating policy. The feedback control path provides a complementary control action to compensate for uncertainties and perturbations about the commanded trajectories. Figure 7.11 shows the ramp response of the FFPU with the addition of the feedforward control. Controller settings are the same as in Table 7.2. There is a meaningful improvement compared to the results in Section 7.3.2. Also, since the variations of the feedback control signals are smaller, the magnitude of the interaction effects between the control loops are less noticeable. An immediate consequence of this effect is that it allows to extend the application range of the PID controllers, thus opening the possibility for process optimization, which implies different operating policies, without retuning the feedback controllers.
Figure 7.11 Response to unit load demand ramp with feedforward control.
7.4.2 Set-point scheduler

The incorporation of the set-point scheduler, described in Chapter 4, further enhances the performance of the FFPU during wide-range operation. The set-point scheduler specifies the set-point trajectories to be followed by the plant, and the neuro-fuzzy feedforward / fuzzy-PID feedback control enforces tracking along the optimally specified set-point patterns, a possibility that was banned using the typical CC due to the drawbacks reported in Section 7.3.2. Figure 7.12 shows the ramp responses for the case reported in Section 7.3.2 using the 2-objectives optimal set-point mappings designed in Section 4.5. Results clearly show an improved response in process operation.

7.4.3 Interaction compensator

Despite the improvements obtained with the incorporation of the set-point scheduler and the feedforward control, the problems of interaction are still present. They still affect the performance of the system; more significantly when the FFPU is operating far away from the tuning point. To alleviate this problem, it was proposed in Chapter 5 to use a 2D scheduled interaction compensator. To show the effect of the compensator, it is incorporated between the PID controllers and the FFPU in the typical CC scheme (no set-point scheduler, no feedforward control). Figures 7.13 through 7.15 show the FFPU step response in the same conditions as in Section 7.3.1. Clearly, the control loop interaction has been reduced. As with the incorporation of the feedforward control, the application range of the PID controllers has been extended.
Figure 7.12 Response to unit load demand ramp with 2 optimization objectives.
Figure 7.13  Response to small set-point power step with interaction compensator.
Figure 7.14  Response to small pressure set-point step with interaction compensator.
Figure 7.15  Response to small level set-point step with interaction compensator.
7.4.4 Fuzzy-PID control

Perhaps the most important consequence of the decoupling effect of the interaction compensator is that it facilitates the controller tuning process. Even though the interaction is not completely eliminated, it is reduced to a good degree that allows independent tuning of each loop, and less retuning is required when all the control loops are closed. In turn, this fact simplifies and makes it easier to find the controller parameters at all the required operating points to implement the fuzzy-PID controllers. Table 7.4 summarizes the selected scheduling points and the corresponding controller parameters for a 40% variation in the process dynamics (described in Section 6.4) for the multiobjective optimal power-pressure mappings with 1 optimization objective, and Table 6.6 summarized the elements of the interaction compensator (described in Section 6.4) along the same operating policy. Figure 7.16 shows the response of the system to set-point steps in the same conditions as in Section 7.3.3 for the 1-objective optimization case.

<table>
<thead>
<tr>
<th>controller parameter</th>
<th>operating point:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>21.5 MW</td>
</tr>
<tr>
<td></td>
<td>70.8 kg/cm²</td>
</tr>
<tr>
<td>$K_p$</td>
<td>0.132</td>
</tr>
<tr>
<td>$K_i$</td>
<td>2.09e-4</td>
</tr>
<tr>
<td>$K_d$</td>
<td>0</td>
</tr>
<tr>
<td>power</td>
<td></td>
</tr>
<tr>
<td>$K_p$</td>
<td>0.28</td>
</tr>
<tr>
<td>$K_i$</td>
<td>5.09e-4</td>
</tr>
<tr>
<td>$K_d$</td>
<td>0</td>
</tr>
<tr>
<td>pressure</td>
<td></td>
</tr>
<tr>
<td>$K_p$</td>
<td>6e-3</td>
</tr>
<tr>
<td>$K_i$</td>
<td>6e-8</td>
</tr>
<tr>
<td>$K_d$</td>
<td>-</td>
</tr>
<tr>
<td>level</td>
<td></td>
</tr>
</tbody>
</table>

TABLE 7.4 PARAMETERS OF PID CONTROLLERS FOR 1-OBJECTIVE CASE
Figure 7.16 Responses to set-point steps at different operating point with fuzzy-PIDs.
7.5 System Integral Tests

Once the role of each component of the ICCS-MP control scheme has been elucidated, it is necessary to evaluate the performance of the whole system. The set of tests in this section are intended to reveal the characteristics of the ICCS-MP control scheme. It is important to highlight that all system tests are commanded through the unit load demand, as would be the case for operation of a FFPU in a totally automated power system. The required unit load demand is assumed to come from upper level unit commitment and economic dispatch applications, which most of the time are remotely located. Whenever possible, comparison is made against the results obtained in Section 7.3 for the typical CC.

7.5.1 Unit load demand step response

As discussed earlier, step responses are not allowed in a FFPU for safety reasons. Even though, the step response of the FFPU is presented here only with demonstration purposes and to show that the ICCS-MP behaves adequately in such cases. Step response tests consist of a step change in the unit load demand that translates into a simultaneous step in the power and pressure set-points. Thus, it should be kept in mind that the unit load demand step is a more demanding test condition than those of independent steps, one at a time, for each set-point as was previously reported. Tests were carried out at three different load levels (40, 80, and 120 MW), in two step sizes (1 MW and 5 MW), and in both directions (up and down) for each one of the operation cases with 1, 2, and 4 optimization objectives for a total of $3 \times 2 \times 2 \times 3 = 36$ test cases. Figure 7.17 shows a typical response under the same conditions of Section 7.3.1 for a small step-up (1 MW) in the unit load demand. Similar behavior was obtained at all load levels and for any one of the operating policies.
Figure 7.17 Response to unit load demand step with ICCS-MP.
7.5.2 Unit load demand ramp response

Basic load-tracking experiments were carried out for the mild conditions in Section 7.3.2, and for a large unit load demand change from 80 MW (50% base load) to 160 MW (100% base load) in 600 seconds (10 min), that is a 50 % change in unit load demand at a rate of 5 %/min, which corresponds to the fastest rate allowable in practice. Other ramps included 6.25% unit load change at 5 %/min load rate, and 50% unit load change at 2.5 %/min load rate. All test were carried out for ramp-up and ramp-down maneuvers for the cases with 1, 2, and 4 optimization objectives, for a total of 4×2×3=24 test cases. Figure 7.18 shows the FFPU response to the fast ramp-up test for the case with 1 optimization objective.

7.5.3 Wide-range cyclic operation

The performance of the FFPU during a cyclic operation regime is tested with an ad-hoc unit load demand pattern that emulates the maneuvers encountered in a typical daily load cycle. The pattern consists of a small ramp-down from 80 MW (50% load) to 40 MW (25% load) at a slow 2 %/min load rate, a fast ramp-up from 40 MW to 160 MW (100% load) at a fast 5 %/min load rate, a medium ramp-down from 160 MW back to 80 MW a moderately fast 3.33 %/min load rate, and constant demand values between the ramps, as introduced in Section 4.53 and shown in Figure 4.13. The same unit load demand pattern is used for the operating scenarios with 1, 2, and 4 optimization objectives that were described in Section 4.5.1. The corresponding trajectories for the power, pressure and level set-points are shown in Figures 4.14, 4.15, and 4.16, respectively. The set-point trajectories were obtained from the unit load demand pattern through the multiobjective optimal set-point mappings designed in Section 4.5.2 and shown in Figures 4.7, 4.8, and 4.9.
Now, Figure 7.19 shows the response of the FFPU to the cyclic unit load demand pattern, including the case with 1 optimization objective. Load-tracking is very good; the differences between the power set-point trajectory and the generated power are almost indistinguishable. Tracking of the pressure set-point trajectories, which vary for each case, is also very good. Some minor oscillations show up after the sharp changes in the set point trajectories, but they lay down well within acceptable tolerances. In all cases good regulation about the zero drum level deviation was obtained with oscillations of very small magnitude. The corresponding behavior of the control signals $u_1$, $u_2$, and $u_3$, is also shown in Figure 7.19, which relate directly to the objective functions $J_2$, $J_3$, and $J_4$, respectively. To have a better appreciation of these results, the values accumulated during the simulation for each one of the four objective functions are provided in Table 7.5, where as is usual for minimization problems, a smaller value indicates better performance, including the negative values for $J_3$ and $J_4$, where the negative values reflect the definition of the objective functions in (4.33c) and (4.33d). The objectives which were not subject to optimization are provided within parenthesis for each case; their values are presented so that all cases can be compared back to back. Unlikely, all objectives improved as the number of objectives increased.

<table>
<thead>
<tr>
<th>Optimization criteria</th>
<th>1-objective</th>
<th>2-objectives</th>
<th>4-objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_1$</td>
<td>153.3</td>
<td>190.6</td>
<td>232.3</td>
</tr>
<tr>
<td>$J_2$</td>
<td>(4152.6)</td>
<td>4069.3</td>
<td>3955.7</td>
</tr>
<tr>
<td>$J_3$</td>
<td>(-5603.1)</td>
<td>(-6098.6)</td>
<td>-7087.5</td>
</tr>
<tr>
<td>$J_4$</td>
<td>(-3892.4)</td>
<td>(-3907.8)</td>
<td>-3947.5</td>
</tr>
</tbody>
</table>
Figure 7.18 Response to fast large unit load demand ramp with ICCS-MP.
Figure 7.19 Response to wide-range unit load demand cycle with ICCS-MP.
7.5.4 Measurement noise

So far, the control system integral test show good performance for the nominal plant under normal operating conditions. In this section the effect of measurement noise on the system performance is taken into account. To this aim white noise is added at the three outputs. The magnitude of the noise is considered to be determined by the quantization effect that would appear due to the analog to digital conversion in an actual implementation. The cyclic tests are repeated with the noise addition. Typical results are provided in Figure 7.20, that shows a satisfactory performance. Also, the process optimization objective values are provided in Table 7.6, which do not significantly deviate from those obtained for the nominal conditions in the previous section.

<table>
<thead>
<tr>
<th>Optimization criteria</th>
<th>1-objective</th>
<th>2-objectives</th>
<th>4-objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_1$</td>
<td>230.2</td>
<td>262.1</td>
<td>304.6</td>
</tr>
<tr>
<td>$J_2$</td>
<td>(4152.0)</td>
<td>4068.3</td>
<td>3954.4</td>
</tr>
<tr>
<td>$J_3$</td>
<td>(-5602.8)</td>
<td>(-6099.2)</td>
<td>-7093.4</td>
</tr>
<tr>
<td>$J_4$</td>
<td>(-3891.0)</td>
<td>(-3905.2)</td>
<td>-3939.0</td>
</tr>
</tbody>
</table>
Figure 7.20  Effect of measurement noise in wide-range cyclic response with ICCS-MP.
7.5.5 Uncertainty on model parameters

The parameters of the valve actuators, modeled as a first order system with rate and position limits, were changed within ample ranges after all design and tuning tasks were finished. In a first case, the actuator nominal gains were modified by a factor in the range [0.5, 2.0] to affect the FFPU steady-state response. In a second case, the time constants of the control valves were affected by a factor also in the range [0.5, 2.0] to affect the FFPU transient response. In both experiments, the performance was observed for a unit load demand ramp from 80 MW to 90 MW at a fast rate of 5 %base-load/min. Figures 7.21 and 7.22 provide the IAE (7.5a) and ISE (7.5c) performance indices, normalized to the value at nominal conditions, at factor values every 0.25 for variations of the control valve gain and time constant in each one of the power ($E$), pressure ($P$), and level ($L$) control loops. These results show good robustness of the control system to uncertainty in the FFPU model parameters since a reasonably large ±15% variation in the parameters can be perfectly tolerated.

7.5.6 Variation of controller parameters

The parameters of the fuzzy-PID controllers were multiplied by a factor in the range [0.5, 2.0], every 0.25, to observe the effect on the FFPU response. The performance was observed for a unit load demand ramp from 80 MW to 90 MW at a fast rate of 5 %base-load/min. Figure 7.23 provides the ISE (7.5c) performance index, normalized to the value at nominal conditions, for variations in each parameter for the controllers in the power ($E$), and pressure ($P$) control loops. In general, results show that changing a parameter may improve the performance in the control loop being considered, but the performance in the other loops will decrease. Fortunately, small parameter variations produce reasonably small changes in performance for most cases.
Figure 7.21 Effect of control valve gain.
Figure 7.22 Effect of control valve time-constant.
Figure 7.23 Effect of controller parameter variations.
7.6 Comments

The performed cyclic tests with the nominal plant show that the ICCS-MP effectively solves the drawbacks of the conventional CC scheme. Cyclic tests under noise measurement confirm the roles of the feedforward and feedback control in the 2 DOF control scheme: feedforward provides for wide-range maneuverability, and feedback compensates for uncertainties and disturbances about the commanded trajectories.

Regarding process optimization, the presented case study shows the methodology to achieve process optimization in a multiobjective sense in a power plant, and the way in which verbal operation requirements were translated into mathematically tractable descriptions in terms of simple objective functions and their relative preferences. Tackling more complex operating objectives (i.e., life extension, and reduction of pollutant emissions) may require a more complete model of the process. In the case presented it should be noted that the objective function $J_1$ accounts for load tracking, while the objective functions $J_2$, $J_3$, and $J_4$ may be related directly to the unit’s heat rate. Thus, covering two of the most important operation requirements.
CHAPTER 8

CONCLUSIONS

8.1 Summary and Conclusions

This dissertation presented a methodology to design a generalized overall unit control system for a fossil-fuel power unit, and developed a minimum prototype to show its feasibility. Basically, the generalized unit control concept extended the capabilities of current coordinated control schemes, and it was presented as the Intelligent Coordinated Control System paradigm. In general, the ICCS paradigm established an open reference framework for the development of large-scale overall power unit control systems.

The design of the ICCS followed a multidisciplinary approach that allowed a systematic treatment of the control system complexity. Due to the scope of this project, the approach amalgamated concepts from three basic disciplines. Process engineering allowed identification of the overall control system goals. Intelligent control engineering concepts were used to identify tasks and to functionally decompose the control system. Software engineering was used to identify and group software agents according to their purpose and knowledge manipulation. The resulting ICCS organization is an open set of functionally grouped agent clusters in a two-level hierarchical system. Agents at the supervisory level provide for self-governing operation characteristics of the control system. Agents at the lower level provide for the reactive behavior functions necessary for real-time control and protection.

The feasibility demonstration of the ICCS followed a software process of the rapid-prototyping type through the realization of a minimum prototype that would exhibit the distinctive characteristics of the ICCS. In this way, the ICCS-MP comprehends a minimum set of functions needed by the power unit to participate in the total automation of power systems. Basically, the ICCS-MP provides the means to achieve optimized
wide-range cyclic operation, by being able to follow any given unit load demand profile issued by upper level economic dispatch and unit commitment agents, and to optimally accommodate an arbitrary number of generally conflicting operating objectives.

The ICCS-MP implemented a two-level hierarchical intelligent hybrid overall unit control system. The supervisory functions include optimization and set-point generation, learning and control tuning, and performance and state monitoring. The direct level consist of a two-degrees-of-freedom multivariable feedforward-feedback control scheme. The core of the system is implemented in three modules: reference governor, feedforward control processor, and feedback control processor. The reference governor generates set-point trajectories for the lower level control loops by solving a multiobjective optimization problem, for which the objective functions and their priorities can be set arbitrarily, in number and form. This approach allows for process optimization, and provides a way to specify the operating policy to accommodate a great diversity of operating scenarios. The proposed feedforward-feedback control scheme is an extension of the general linear single-input-single-output feedback control scheme, with both reference feedforward and disturbance feedforward actions, to the nonlinear multivariable case. The feedforward control processor is implemented using a set of multi-input-single-output fuzzy inference systems designed from plant input-output data using a neural network paradigm. This approach provides the control system with off-line learning capabilities to attain process optimization under changing operating conditions. The feedback control path is implemented as a PID-based decentralized (multiloop) control scheme with a loop interaction compensator. The compensator is equivalent to a disturbance feedforward compensator and is designed using the relative gain array technique. Both the control algorithms and the compensator are first order Sugeno-type fuzzy inference systems, scheduled in two dimensions (power and pressure) to achieve satisfactory disturbance rejection and uncertainty compensation during wide-range operation. The process operating window is partitioned to take into account the process nonlinear characteristics, and tuning is carried out by a genetic algorithm at the points of interest in the partitions.
The performance of the ICCS-MP was demonstrated through extensive simulation experiments. Results showed the feasibility of the proposed ICCS paradigm, that is, an open purposeful self-governing overall unit control system for a FFPU can be systematically designed, built and upgraded to effectively satisfy arbitrary operation conditions. Remarkably, the ICCS paradigm provided a convenient conceptual framework for the integration of diverse applications, making use of the best features that either algorithmic or heuristic techniques have to offer, while keeping the complexity of a large control system within manageable bounds.

8.2 Contributions

The major objective of this dissertation was the development of the conceptual model, as an open reference framework, for the design and realization of large-scale complex control systems for power units. Technologically, the objectives included the provision of the methods or procedures for a first realization of the proposed conceptual model, to obtain an assessment of its feasibility. Moreover, the achievement of these objectives allows to point out further development and refinement needs for a new generation of intelligent hybrid control systems, and will eventually lead to their actual application in power unit. From this perspective, the contributions of this dissertation include:

- The identification and conceptualization of the overall power unit control problem at the core of any integral automation scheme for fossil-fuel power units. The control problem is recast as the wide-range multiobjective optimal continuous control of a (high-risk) nonlinear multivariable (large-scale) process with highly interactive dynamics, subject to multiple everyday-changing performance requirements and tight constraints. Also, the proposed overall unit control system contributes towards the development of intelligent hybrid control systems that incorporate heuristic and
analytical techniques in a consistent symbiosis that makes use of the best characteristics that each technique has to offer [Garduno and Lee 2000d, Garduno and Lee 2000f].

- The proposition of the ICCS multiagent-based paradigm as a fundamental premise to design large scale complex intelligent control systems for power units. The resultant ICCS organization is that of an open set of functionally grouped agent clusters in a two-level hierarchical structure. The proposition also includes the ICCS-MP control scheme as an extension to current coordinated control schemes in FFPUs. [Garduno and Lee 1999d, Garduno and Lee 2000g, Garduno and Lee 2001].

- A method for wide-range multiobjective process optimization through set-point scheduling, which includes a way to specify the desired operating strategy in terms of simple objective functions and their normalized relative preferences. This approach allows accommodation of operating scenarios characterized by multiple operating requirements [Garduno and Lee 1999c, Garduno and Lee 1999f, Garduno and Lee 2000e].

- A method to achieve wide-range operation of a power unit through feedforward neuro-fuzzy control. The feedforward controller is an open system that can be designed from input-output steady-state process data using a neural network training algorithm that provides the ICCS-MP with learning capabilities [Garduno and Lee 1999b, Garduno and Lee 1999e].

- A method for multiloop control of a power unit for wide-range operation using 2D scheduling of a class of multimode fuzzy-PID controllers and feedforward interaction compensation. The design of the feedback controller (fuzzy-PID and compensator) does not require a model of the process and its tuning procedure can be automated [Garduno and Lee 2000a, Garduno and Lee 2000c, Garduno and Lee 2000h].
It is important to mention here that some additional research products, although directly related as either antecedents or variations of some of the results previously mentioned, were not incorporated into this dissertation because they were superseded by later or improved developments that showed better consistency with respect to the whole control system philosophy. These developments include a supervisory fuzzy set-point scheduler [Garduno and Lee 1997], a fuzzy coordinated control strategy for power units [Garduno and Lee 1999a], and a fuzzy-PID controller with reduced rule sets [Garduno and Lee 2000b].

8.3 Future Research

There are many challenges and opportunities along the ICCS research line. Among the most immediate and viable are the following:

- Research on the ICCS paradigm and on the ICCS-MP control scheme will be continued through the development of an Intelligent Hybrid Overall Control System for Turbogas Units. This project will be carried out as an international joint research project between the Power Systems Laboratory at Penn State (USA) and the Instrumentation and Control Unit at the Electric Research Institute (Mexico). There will be further exploration, simulation and analytical, on the stability and performance robustness characteristics of the ICCS-MP control scheme, since it is critically required at turbogas units to increase their efficiency for wide-range operation, including operation very close or in the unstable surge zone. The fuzzy-PID control scheme will be evaluated to extend the operating load region, and the inverse decoupling compensation scheme will be evaluated to extend the stable operating zone at any given load.
• The set-point scheduler is model-based. The adequacy of the set-points it generates depends on the validity and accuracy of the model being used. As for any model-based strategy this is a difficult issue, which gets even more complicated by the wide-range operation requirements posed on the power unit. It is suggested that a fuzzy multiobjective set-point scheduler be used instead, with no requirement of a process model. This second generation set-point scheduler may implement a multiobjective evolutionary optimization strategy by combining the experimental sequential simplex optimization method [Walters, et al. 1991] and fuzzy multiobjective decision making theory [Lai and Hwang 1994].

• A big step toward real-world implementation can be achieved by incorporating the required sequence control and protection functions into the extended coordinated control strategy of the ICCS-MP, so that all operation sequences (start-up, shutdown, etc.) may be executed automatically. Integration of continuous control and sequence control may be done using the theoretical framework of hybrid systems and finite state automatons [Alvarez, et al. 1999]. Proceeding in this way, the direct control agent cluster in the ICCS functional block diagram will be completely incorporated. A first attempt in this direction has already been made for a 1 MW turbogas power unit [Ramirez 1996].

• From the software engineering point of view, this dissertation only pointed out, and demonstrated through rapid prototyping, the direction in which new, more complex, control systems for power units should be designed and implemented. The work in this dissertation is only a first approximation to the design of an intelligent multiagent system. The architecture of the system has been defined, and the minimum prototype applications are already developed. Nevertheless, to achieve the true philosophy of software agency, the design and implementation of agents should be further refined. In addition, the implementation of a second generation prototype should be pursued using more adequate software tools, some of which need to be developed. For
instance, a PC-based development platform for intelligent multiagent systems. This platform should comprehend at least a library of low-level functions to provide for the functionality of the platform, an agent description language to specify the agents and to translate them into executable code, and a language for common knowledge representation across all applications to allow information exchange among agents.


---

In this list, we are referring to Gardner R. and K.Y. Lee's contributions to the field of power systems control, specifically focusing on the use of fuzzy logic and intelligent systems to optimize and control power units. Their work spans a range of conferences and publications, showcasing advancements in the application of fuzzy logic and control strategies to enhance the performance and efficiency of power systems.


VITA

Raul Garduno-Ramirez was born in Cuernavaca, Mexico. He received the B.S.E.E. degree from the National Polytechnic Institute (IPN), D.F., Mexico, in 1985, and was nominated among Mexico’s Best Students. He received the M.S. degree in Electrical Engineering from the IPN Advanced Research Centre, D.F., Mexico, in 1987. During 1986 he stayed at the National Mechanical Laboratory, Tsukuba, Japan, as a JICA Scholar researching on expert control and robotics. From 1987 through 1995 he stayed at the Electric Research Institute (IIE), Cuernavaca, Mexico, where he was involved in the development of control systems for power plants as both research engineer and project leader. Also during that period, he was elected a National Researcher, and hold a joint position at the National Centre for Research and Technology Development (Cenidet), Cuernavaca, Mexico, lecturing on controls and robotics. Then, as a Fulbright Scholar, he came to the Pennsylvania State University to pursue a Ph.D. degree in Electrical Engineering, doing research in power plant control. His current interests include control software development, intelligent control, playing tennis, sharing life with his family, and would like to write something interesting and useful …